



Titre: Design et tests in vivo d'un microstimulateur urinaire sélectif
Title: implantable

Auteur: Stéphane Boyer
Author:

Date: 2003

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Boyer, S. (2003). Design et tests in vivo d'un microstimulateur urinaire sélectif implantable [Master's thesis, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/7114/>

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

URL de PolyPublie: <https://publications.polymtl.ca/7114/>
PolyPublie URL:

Directeurs de recherche: Mohamad Sawan
Advisors:

Programme: Unspecified
Program:

In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

UNIVERSITÉ DE MONTRÉAL

DESIGN ET TESTS IN VIVO D'UN MICROSTIMULATEUR URINAIRE
SÉLECTIF IMPLANTABLE.

STÉPHANE BOYER
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE)
JUILLET 2003



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

ISBN: 0-612-86386-7

Our file Notre référence

ISBN: 0-612-86386-7

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.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

DESIGN ET TESTS IN VIVO D'UN MICROSTIMULATEUR URINAIRE
SÉLECTIF IMPLANTABLE.

présenté par: BOYER Stéphane

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. KHOUAS Abdelhakim, Ph.D., membre

À Fernande et Hervé.

REMERCIEMENTS

Je tiens d'abord à remercier Mohamad Sawan, professeur à l'École Polytechnique et directeur du laboratoire PolySTIM, pour m'avoir permis de contribuer à un projet aussi gratifiant, pour son support moral et ses encouragements.

Je remercie également les membres de l'équipe PolySTIM, et plus particulièrement Luc Romain, Jean-François Harvey et Simon Robin, pour les nombreuses fois où ils ont su me faire voir un problème sous un angle différent et surtout pour avoir pavé la voie de l'inconnu devant moi.

Je salue aussi le personnel du centre de ressource animale (ARC) de l'Université McGill, sous la supervision du Docteur Mostafa M. Elhilali, pour leur patience et leur professionnalisme lors des expérimentations.

Pour terminer, je remercie la Fondation canadienne du rein et le Conseil de recherche en sciences naturelles et génie du Canada pour le soutien financier au projet.

RÉSUMÉ

Parmi les traitements disponibles pour récupérer les fonctions urinaires chez les paraplégiques, la stimulation électrique des racines sacrées est l'une des avenues les plus prometteuses. Toutefois, la stimulation de ce site provoque la contraction simultanée de muscle entourant la vessie (détrusor) et du sphincter urétral externe, inhibant la miction et causant une pression anormalement élevée dans la vessie. La stimulation sélective à l'aide de deux trains d'impulsion de fréquence différente est l'une des techniques utilisées pour réduire la dyssynergie et favoriser la miction.

L'équipe de recherche PolySTIM met au point depuis plus d'une décennie différents stimulateurs implantables dédiés à la récupération des fonctions urinaires. Un nouveau microstimulateur intégré a été réalisé dans le cadre de cette maîtrise à l'aide d'un procédé de fabrication de $0,35\ \mu\text{m}$. Le circuit mixte, intégrant un module numérique de génération de signaux et une source de courant bidirectionnel de 5 bits, permet de produire la forme d'onde nécessaire à la stimulation sélective.

L'article «Implantable Selective Stimulator to Improve Bladder Voiding: Design and Chronic Experiments in Dogs», publié dans «IEEE Transactions On Rehabilitation Engineering», présente le système de stimulation, le protocole et les résultats d'essais en phase chronique réalisés sur 8 chiens. Les résultats montrent que la stimulation sélective des racines sacrées favorise la miction en inhibant la contraction du sphincter externe tout en permettant l'excitation du détrusor. Elle augmente

de plus de 50% l'évacuation d'urine de la vessie par rapport à une technique de stimulation basse fréquence uniquement.

Enfin, un stimulateur externe et portable a également été réalisé à l'aide du microcontrôleur EPSON E0C88316. Ce système, utilisant un écran LCD et un clavier, offre une interface utilisateur évoluée permettant de modifier chacun des paramètres de stimulation, de conserver plusieurs ensembles de paramètres précédemment définis et de les récupérer au besoin. L'étage de sortie utilisé pour ce design est le même que celui présenté avec le nouveau stimulateur intégré et permet le contrôle du courant de stimulation. L'uniformisation des appareils utilisés tout au long du protocole devrait permettre de tirer des conclusions plus justes.

ABSTRACT

Among the available treatment to retrieve bladder functions for paraplegics, electrical stimulation of the sacral roots seems to be one of the most promising solutions. However, neurostimulation of this site provokes a simultaneous contraction of the bladder muscle and the external urethral sphincter, preventing micturition and causing abnormally high bladder pressure. Selective stimulation using two series of impulsions at different frequencies is one of the techniques used to reduce the detrusor-sphincter dyssynergia and assist micturition.

The PolySTIM research team designs, since more than a decade, different types of implantable electrical stimulators used to retrieve bladder functions. In this master thesis, a new integrated microstimulator was carried out using a $0.35\ \mu\text{m}$ fabrication process. The mixed circuit, integrating digital logic for waveform generation and an bidirectional 5-bit current source can produce the necessary waveform for selective stimulation.

The paper "Implantable Selective Stimulator to Improve Bladder Voiding: Design and Chronic Experiments in Dogs", published in "IEEE Transactions On Rehabilitation Engineering", presents the medical protocol and results of chronic studies on 8 mongrel dogs. Results show that selective stimulation of sacral roots promotes micturition by preventing external sphincter contraction and having little effect on detrusor contraction. This technique increases micturition volume

by more than 50% when compared with standard low-frequency only stimulation technique.

Finally, an external and portable stimulator has also been realized with the EPSON E0C88316 microcontroller. This system, fitted with an liquid crystal display (LCD) and a keyboard, offers a friendly user interface with which every stimulation parameter can be viewed and modified and, parameters' set can be saved and retrieved when necessary. The output stage of this design is the same as the implantable stimulator described earlier and enables to control the stimulation current. The uniformisation of devices' output stage used along each step of the protocol should allow to draw more accurate conclusions.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES TABLEAUX	xiv
LISTE DES FIGURES	xv
LISTE DES NOTATIONS ET DES SYMBOLES	xviii
LISTE DES ANNEXES	xxi
INTRODUCTION	1
CHAPITRE 1: LE SYSTÈME URINAIRE ET LA STIMULATION	
ÉLECTRIQUE	4
1.1 Introduction	4
1.2 L'appareil urinaire	4
1.2.1 La partie inférieure de l'appareil urinaire	5

1.2.2	Les tissus musculaires de la vessie	6
1.3	Les voies neurales	6
1.3.1	Les voies sensibles	7
1.3.2	Les voies motrices de l'appareil urinaire	8
1.3.3	Les centres nerveux réflexes	8
1.3.4	Les centres nerveux corticaux	9
1.4	La miction normale	9
1.5	Les effets d'un traumatisme	11
1.6	La stimulation neuromusculaire	13
1.6.1	Les sites de stimulation	13
1.6.2	Les techniques de stimulation des racines sacrées	15
1.7	Conclusion	18

CHAPITRE 2: LE SYSTÈME DE STIMULATION NEUROMUS- CULAIRE 19

2.1	Introduction	19
2.2	Les systèmes commerciaux	19
2.3	Aperçu sur l'évolution de stimulateurs offerts par PolySTIM	20
2.4	Les contrôleurs externes	24
2.5	Les électrodes	26
2.6	Conclusion	27

CHAPITRE 3: SIMULATEUR IMPLANTABLE ET VALIDATION

DE LA TECHNIQUE DE STIMULATION SÉLEC-

TIVE EN PHASE CHRONIQUE 29

3.1	Introduction à l'article publié dans «IEEE Transactions on Rehabil- itation Engineering»	29
3.2	Article «Implantable Selective Stimulator to Improve Bladder Void- ing: Design and Chronic Experiments in Dogs»	30
3.2.1	Abstract	30
3.2.2	Introduction	31
3.2.3	Description of the Stimulation System	37
3.2.4	Experimental Protocol	42
3.2.5	Results	45
3.2.6	Discussion	51
3.3	Conclusion	53

CHAPITRE 4: RÉALISATION ET TEST D'UN STIMULATEUR

IMPLANTABLE INTÉGRÉ 55

4.1	Introduction	55
4.2	Design	57
4.2.1	Architecture	57
4.2.2	Alimentation de l'implant et réception RF	58
4.2.3	Module numérique de réception et de contrôle	58

4.2.4	Étage analogique de sortie	59
4.3	Vérification, synthèse et dessin des masques	62
4.4	Test du circuit intégré	66
4.5	Conclusion	71

CHAPITRE 5: RÉALISATION D'UN STIMULATEUR EXTERNE

	PORTABLE	73
5.1	Introduction	73
5.2	Architecture générale	73
5.3	Interface utilisateur	76
5.3.1	Modification des paramètres	77
5.3.2	Gestion des espaces mémoires	78
5.3.3	Stimulation neuromusculaire	78
5.3.4	Sous-routine de rafraîchissement de l'affichage	79
5.4	Génération des stimuli et étage analogique de sortie	81
5.5	Vérification et tests	82
5.6	Conclusion	82

	DISCUSSION GÉNÉRALE ET CONCLUSION	84
--	--	-----------

	BIBLIOGRAPHIE	89
--	--------------------------------	-----------

LISTE DES TABLEAUX

1.1	Résumé des techniques de stimulation des racines sacrées dédiées à la vidange de la vessie.	18
2.1	Résumé des caractéristiques des neurostimulateurs implantables. . .	24
2.2	Résumé des caractéristiques des contrôleurs externes.	26
3.1	Frequently employed parameters: low-frequency only and selective stimulations.	44
3.2	Voided and residual urine quantities in eight chronic dogs.	48
4.1	Plage disponible pour les paramètres de stimulation de l'implant. .	56
4.2	Comparaison des caractéristiques des sources développées par St-Amand (1995), Bourret (1999) et de la nouvelle version proposée. .	70

LISTE DES FIGURES

1.1	Représentation simplifiée de l'appareil urinaire.	5
1.2	Les voies neurales du système urinaire.	7
2.1	Diagramme simplifié du neurostimulateur introduit par Sawan. . . .	21
2.2	Diagramme simplifié du neurostimulateur introduit par Arabi. . . .	22
2.3	Schéma bloc du neurostimulateur introduit par Robin.	23
2.4	Électrode à gaine cylindrique avec armature en alliage à mémoire de forme réalisée par Crampon.	27
3.1	Possible stimulation sites and implantable stimulator position for high-frequency inhibition.	31
3.2	Typical selective stimulation waveform intended to block the sphinc- ter activation during bladder contraction (waveforms are not to scale; meaning of abbreviations are given in Table 3.1).	38
3.3	Block-diagram of the whole proposed selective stimulator including the external controller, the implant and the electromagnetic link between both parts.	39
3.4	Simplified block diagram of the output stage of the implant including an analog switch array to deliver fully balanced bipolar stimuli. . .	41

3.5	X-ray views of the bladder: a) external urethral sphincter (EUS) contracted, b) EUS relaxes with selective stimulation and, c) residual urine volume (shown by arrows).	47
3.6	Residual volume for three stimulation steps: low-frequency only, selective and post-selective (for complementary information, see Table 3.2).	49
3.7	Intravesical and intraurethral pressures and EMG of the sphincter: a) lower intraurethral pressure allows micturition; b) reduced EMG coincides with lower intraurethral pressure.	50
4.1	Forme du signal généré par le stimulateur implantable.	56
4.2	Schéma simplifié du stimulateur neuromusculaire implantable. . . .	57
4.3	Schéma simplifié du circuit intégré dédié.	59
4.4	Schéma électrique de l'étage analogique de sortie (CNA et étage de sortie bipolaire).	60
4.5	Géométrie de type centre de masse commun utilisée pour le CNA. . .	62
4.6	Dessin des masques de la source de courant.	63
4.7	Courants de sortie mesurés en fonction de différentes tensions de référence.	68
4.8	Non-linéarités différentielle et intégrale du convertisseur pour une tension de référence de 0,93 volt.	69

4.9	Non-linéarités différentielle et intégrale du convertisseur pour une tension de référence de 1,2 volt.	69
5.1	Photo du prototype du stimulateur externe portable.	74
5.2	Architecture générale du stimulateur externe.	75
5.3	Algorithmes du programme principal du stimulateur externe et de la routine d'exception d'entrée au clavier.	76
5.4	Lorsque l'espace d'affichage est insuffisant, la navigation se fait à l'aide des flèches.	77
5.5	Algorithme de la sous-routine de rafraîchissement de l'affichage de l'écran LCD.	80
II.1	Schéma électrique du stimulateur neuromusculaire externe et portable.	121

LISTE DES NOTATIONS ET DES SYMBOLES

ASD:	Analyse statique de délais
ASK:	Amplitude Shift Keying
BiCMOS:	Bipolar and CMOS
CI:	Circuit intégré
CMOS:	Complementary Metal-Oxide Semiconductor
CNA:	Convertisseur numérique à analogique
CRSNG:	Conseil de recherches en sciences naturelles et en génie du Canada
DAC:	Digital to Analog Converter
DIP:	Dual In-line Package
DSD:	Detrusor-Sphincter Dyssynergia
EPROM:	Electrically Erasable Programmable Read-Only Memory
EMG:	Electromyograph
EUS:	External Urethral Sphincter
FES:	Functional Electrical Stimulation
FPGA:	Field Programmable Gate Array
FSM:	Finite State Machine
IC:	Integrated circuit
IEEE:	Institute of Electrical and Electronics Engineers
IVU:	Intravenous Urography

KFC:	Fondation canadienne du rein (Kidney Foundation of Canada)
LCD:	Liquid Crystal Display
LED:	Light Emitting Diode
LSB:	Least Significant Bit
MSB:	Most Significant Bit
MSN:	Microstimulateur neural sélectif
NLB:	Neurostimulateur à lien bidirectionnel
NLD:	Non-linéarité différentielle
NLI:	Non-linéarité intégrale
NSERC:	Natural Sciences and Engineering Research Council of Canada
NSP:	Neurostimulateur sélectif et permanent
OE:	Onde excitante
OI:	Onde inhibitrice
PC:	Ordinateur personnel (Personal Computer)
PCB:	Printed Circuit Board
PEPU:	Processeur externe pour une prothèse urinaire
RF:	Radio-fréquence (Radio-frequency)
SCM:	Société canadienne de micro-électronique
TSMC:	Taiwan Semiconductor Manufacturing Company
VCUG:	Voiding Cystourethrogram
VHDL:	VHSIC Hardware Description Language

VHSIC: Very High Speed Integrated Circuit

LISTE DES ANNEXES

Annexe I:	Code VHDL du stimulateur implantable et de son	
	environnement de vérification	97
I.1	Code VHDL du «test bench»	97
I.2	Code VHDL du stimulateur implantable	100
Annexe II:	Schéma électrique et code assembleur du stimulateur	
	externe portable	121
II.1	Schéma électrique	121
II.2	Code assembleur du stimulateur externe	122

INTRODUCTION

Les forces électriques sont essentielles à la vie, que ce soit au niveau du métabolisme de la cellule que de la conscience humaine engendrée par l'activité du cerveau. En introduisant un courant produit de façon artificielle, il est possible de modifier ou d'activer ces forces électriques. Ce type d'interaction entre l'énergie électrique et le corps humain est connu depuis plus de 2000 ans ^[30]. L'utilisation des décharges électriques de la torpille (poisson) a été rapportée aussi tôt qu'en l'an 46 pour le traitement de la douleur, mais c'est en 1791 qu'une étude quantitative fut publiée par Luigi Galvani sur le phénomène de la stimulation électrique ^[12]. Galvani a observé la contraction des muscles d'une grenouille en touchant les nerfs avec une paire de ciseaux pendant un orage électrique et il s'est par la suite assuré que ce phénomène était relié à l'électricité. Au cours des quatre dernières décennies, les progrès des procédés de fabrication en électronique ont permis le développement de différents appareils dédiés à la récupération de fonctions motrices, viscérales et sensorielles. La science médicale utilise désormais les stimulateurs cardiaques pour régulariser le rythme du coeur, les implants cochléaires pour récupérer l'ouïe et d'autres stimulateurs électriques pour réduire la douleur, aider à la respiration et contrôler la fermeture de la main. Cependant, plusieurs dysfonctions, comme la perte des fonctions urinaires (évacuation et rétention), ne sont traitées que par médication ou par voies chirurgicales. La perte du contrôle volontaire de la vessie

peut survenir, entre autres, à la suite d'une blessure à la colonne vertébrale au niveau D12 ou supérieur. Pour récupérer les fonctions urinaires, différents sites de stimulation électrique sont possibles mais ce sont les racines sacrées S2 à S4 qui montrent les résultats les plus prometteurs ^[32]. Les racines sacrées contiennent des fibres nerveuses qui contrôlent (innervent) à la fois le muscle vésical (detrusor) et le sphincter urétral externe. Ainsi, une excitation de ces nerfs provoquent une contraction simultanée du detrusor qui augmentent la pression intravésicale, et du sphincter qui réduit ou inhibe l'évacuation d'urine (miction). Plusieurs solutions sont possibles pour contourner ce problème avec de nombreux avantages et inconvénients. La solution proposée par notre équipe consiste à stimuler les racines sacrées avec un signal composé de deux trains d'impulsions de courant bipolaires, un train basse fréquence - haute amplitude pour contracter la vessie, et un train haute fréquence - basse amplitude pour inhiber la fermeture du sphincter.

Du point de vue matériel, les stimulateurs électriques disponibles commercialement ne sont pas en mesure de produire les signaux nécessaires à la technique de stimulation sélective. Ils sont limités en fréquence, possèdent un nombre de canaux restreints et la forme du stimulus qu'il génère est prédéterminée. Un prototype de stimulateur implantable qui génère les stimuli nécessaires a été réalisé par Robin ^[34]. Ce stimulateur, réalisé sur circuit imprimé, a permis de valider la méthode de stimulation lors d'essais *in vivo*, de déterminer les gammes de paramètres utiles et de vérifier la fiabilité du système.

Suite aux succès du prototype et devant le besoin d'outils plus performants et versatiles, deux nouveaux stimulateurs ont été réalisés. Tout d'abord, une version miniaturisée de l'implant a permis de réduire la consommation de puissance et, éventuellement, de continuer la validation de la méthode de stimulation sur des petits animaux (rats, lapins). Ensuite, un stimulateur externe et portable offre de modifier ponctuellement les paramètres de stimulation à travers une interface usager simple à utiliser.

Le présent mémoire traite, dans le chapitre 1, du système urinaire et des effets d'un traumatisme ainsi que de la stimulation électrique dans le but de provoquer la miction. Le chapitre 2 offre une revue des systèmes de stimulation neuromusculaire commerciaux ainsi que ceux réalisés par l'équipe de recherche PolySTIM. Ce mémoire comprend également l'article «Implantable Selective Stimulator to Improve Bladder Voiding: Design and Chronic Experiments in Dogs» publié dans la revue «IEEE Transactions On Rehabilitation Engineering». L'article est intégré au chapitre 3 et traite, entre autres, des tests fonctionnels et des résultats expérimentaux d'essais en phase chronique. La réalisation, la vérification et le test du stimulateur implantable intégré sont introduits au chapitre 3 mais détaillés dans le chapitre 4. La réalisation d'un stimulateur externe portable, quand à elle, est discutée au chapitre 5 et est suivie des conclusions et recommandations.

CHAPITRE 1

LE SYSTÈME URINAIRE ET LA STIMULATION ÉLECTRIQUE

1.1 Introduction

La connaissance du système urinaire est essentielle pour mener à terme un projet visant la récupération des fonctions vésicales. Pour concevoir un système de stimulation neuromusculaire dédié à évacuer l'urine de la vessie, il est nécessaire d'approfondir les notions de base sur l'anatomie et la physiologie de l'appareil urinaire ainsi que sur l'organisation du système nerveux. En plus de faire une revue rapide de ces notions, ce chapitre traite également des sites et techniques de stimulation électrique provoquant la miction.

1.2 L'appareil urinaire

Le système urinaire regroupe les organes qui aident à maintenir le corps en état d'équilibre principalement par la filtration du sang, la fabrication, l'accumulation et l'évacuation de l'urine. Tel que montré à la figure 1.1, dans la partie supérieure, les reins éliminent les déchets et toxines accumulés dans le sang et les acheminent avec le surplus d'eau vers la vessie par les canaux collecteurs d'urines (bassinets) qui se prolongent en deux fins canaux (uretères) jusqu'à la vessie.

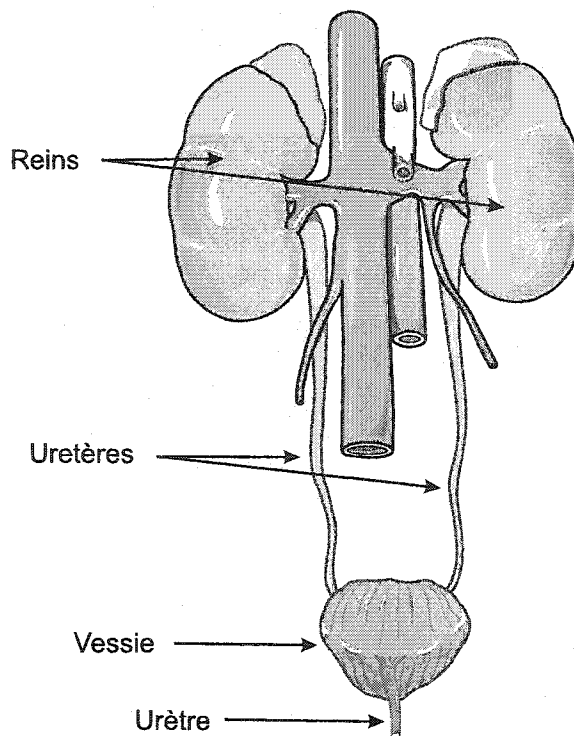


Figure 1.1: Représentation simplifiée de l'appareil urinaire.

1.2.1 La partie inférieure de l'appareil urinaire

Les uretères rejoignent la vessie à sa base en traversant obliquement la paroi musculaire (hiatus urétéral). L'urine est évacuée par l'urètre, le troisième orifice pouvant être vu de l'intérieur de la paroi vésicale; il est le lien entre la vessie et le méat urinaire (orifice externe de l'urètre). La résistance urétrale, lors de la miction, est naturellement plus importante chez l'homme dont l'urètre, plus long que celui de la femme, passe à travers la prostate et le pénis. Les points d'encrage des uretères dans la vessie ainsi que l'urètre forme l'entité topographique appelée «trigone». Cette partie de la vessie change de morphologie pendant l'évacuation de l'urine et il en sera question plus loin dans la description de la miction.

1.2.2 Les tissus musculaires de la vessie

La rétention et l'évacuation de l'urine nécessitent principalement la coordination de trois muscles: le détrusor et les sphincters lisse et strié. Le détrusor représente les différentes couches musculaire de la vessie. Il est composé de fibres musculaires lisses organisées en trois couches réalisant une structure bien adaptée à la fonction contractile de la vessie. Le sphincter lisse est un prolongement du détrusor autour du col de la vessie et de l'urètre. Il est également composé de fibres lisses plus résistantes à la fatigue. Le sphincter strié entoure le sphincter lisse et prédomine 2 à 3 centimètres au-dessous du col. Ce dernier assure le contrôle de la continence active (i.e. consciente) lorsque la vessie approche sa pleine capacité et que le besoin d'uriner se fait ressentir. D'autres muscles peuvent jouer un rôle mineur dans le processus de la miction mais leur intérêt est réduit chez les paraplégiques qui, en général, n'ont pas le contrôle volontaire de ces muscles.

1.3 Les voies neurales

Le contrôle des tissus musculaires est assuré par les centres nerveux via les voies sensibles et motrices. Pour produire un stimulus électrique capable de provoquer la miction, il est essentiel d'élaborer sur ces systèmes.

1.3.1 Les voies sensitives

Les récepteurs sensitifs situés dans les viscères transmettent leurs influx aux centres nerveux par le biais des voies afférentes (sensitives). Les sensations sont divisées en deux catégories, la sensibilité extéroceptive regroupe les sensations de douleur, de température et de toucher, tandis que la sensibilité proprioceptive regroupe les sensations d'étirement et de tension des tissus musculaires. La majorité des neurones sensitifs atteignent la partie sacrée de la moelle épinière par l'intermédiaire des nerfs érecteurs et des nerfs honteux internes (figure 1.2). Une partie des neurones gagnent la moelle épinière au niveau thoraco-lombaire par l'intermédiaire des nerfs hypogastriques. Ces neurones afférents concernent surtout la sensibilité extéroceptive du trigone et ne seront pas abordés puisque nous nous concentrerons sur les fonctions urinaires de base.

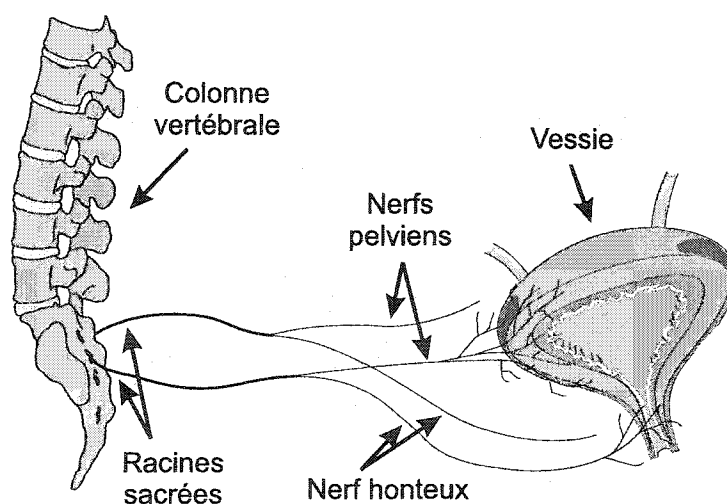


Figure 1.2: Les voies neurales du système urinaire.

1.3.2 Les voies motrices de l'appareil urinaire

Les influx nerveux contrôlant les muscles de l'appareil urinaire sont produits par le système nerveux centrale et acheminés aux tissus musculaires par le biais des voies efférentes (motrices) en empruntant l'innervation somatique ou l'innervation végétative (autonome). L'innervation végétative règle le fonctionnement des viscères et est composée de deux systèmes régulateurs, le sympathique et le parasympathique. Quand à l'innervation somatique, elle règle le fonctionnement moteur du corps. Le détrusor est approvisionné par les fibres parasympathiques via les nerfs pelviens tandis que le sphincter strié est innervé par les fibres somatiques via les nerfs honteux à partir de S2, S3 et S4. Ces conclusions ne font pas l'unanimité au sein de la communauté médicale et plusieurs travaux arrivent à des conclusions différentes ^[14] mais, du point de vue de la stimulation électrique, elles n'influencent pas l'analyse des résultats présentée plus loin.

1.3.3 Les centres nerveux réflexes

Les centres réflexes se situent au point de rencontre d'un neurone sensitif et d'un neurone moteur. Ils organisent une activité réflexe et échappent au contrôle de la volonté. Ils restent cependant sous l'influence de l'ensemble du système nerveux et la libération de cette influence modifie définitivement leur fonctionnement. Le centre réflexe qui contrôle la contraction du sphincter en fonction du volume vésical est le principal responsable de la continence passive (i.e. la fermeture inconsciente

du sphincter lisse). Le centre réflexe reçoit les influx des neurones de sensibilité proprioceptive (tension et étirement) situés dans la paroi vésicale et commande la fermeture du sphincter lisse pour assurer la continence lorsque la vessie se remplit.

1.3.4 Les centres nerveux corticaux

Le contrôle de la vessie implique plusieurs centres nerveux situés dans le cerveau et au niveau de la colonne vertébrale. Les centres corticaux régissent les actions volontaires ou instinctives en percevant les informations venues de la périphérie et en émettant des ordres exécutés par les centres subalternes. Ils sont la base de l'élaboration de la pensée et la décision d'accomplir une action dépend à la fois de la perception des stimuli mais également de la personnalité, des habitudes ou de l'état d'esprit du sujet.

Des notions supplémentaires sur la propagation des potentiels d'action et la transmission des influx nerveux ont été rapportés par Arabi ^[3].

1.4 La miction normale

Maintenant que les organes, nerfs et tissus responsables de la miction ont été identifiés, nous pouvons décrire le processus normal d'évacuation de l'urine. Cet exercice permet de bien saisir la complexité de la miction et montre l'étendue de la tâche à accomplir.

Comme mentionné précédemment, les reins sont responsables du remplissage

graduel de la vessie. Au début du remplissage, la pression vésicale reste basse et la pression urétrale, nettement supérieure, s'élève graduellement par la contraction du sphincter lisse qui assure la continence. Cet automatisme vésico-sphinctérien constitue la continence passive et ne fait pas appel à la conscience ^[14]. Le point de départ de ce réflexe sympathique est la stimulation des récepteurs sensitifs situés dans la paroi vésicale. Les neurones afférents gagnent la moelle épinière par les nerfs érecteurs (pelviens) tandis que la réponse gagne la vessie et l'urètre par les nerfs hypogastriques (honteux). Ce réflexe n'intervient significativement qu'en fin de remplissage ^[31].

Le besoin d'uriner se fait sentir lorsque le volume d'urine atteint entre 250 et 350 ml et que la vessie est suffisamment distendue. Le sphincter lisse abandonne à ce moment la forme d'un disque horizontal pour prendre la forme d'un entonnoir dont l'effet est de redistribuer les forces et de favoriser la miction. Le sphincter strié doit alors être consciemment resserré pour éviter l'écoulement d'urine. La contraction du sphincter strié dépend du centre somatique sacré sous le contrôle des centres corticaux. L'innervation afférente autant que l'efférente emprunte les nerfs honteux internes pour rejoindre les neurones sensitifs et les muscles.

Au moment opportun, les sphincters lisse et strié sont relâchés par une décharge parasympathique. La pression urétrale diminue tandis que la pression vésicale augmente par la contraction du détrusor, faisant place à la miction. Le détrusor se contracte par la réduction d'influx inhibiteurs laissant ainsi agir les influx facili-

tateurs dont la source principale est la stimulation des récepteurs de tension du détrusor. Lorsque la contraction est amorcée, le phénomène s'amplifie de façon automatique et augmente le débit urinaire.

1.5 Les effets d'un traumatisme

Une lésion à la colonne vertébrale au niveau cervical ou dorsal peut provoquer plusieurs modifications irréversibles de l'appareil urinaire. La propagation des influx nerveux entre les centres sacrés et corticaux étant réduite, un déséquilibre entre les influx inhibiteurs et facilitateurs apparaît. La vessie autonome ne dépend plus des centres corticaux mais uniquement des centres réflexes qui régulent ses activités.

Un premier effet d'une lésion à la colonne vertébrale est la perte de sensation de plénitude de la vessie. Une personne paraplégique ne reçoit plus les informations concernant la quantité d'urine présente dans sa vessie. Ainsi, lorsque la vessie approche sa capacité maximale, l'individu ne peut entreprendre les actions nécessaires à l'évacuation de l'urine.

Un tel traumatisme provoque également la perte du contrôle volontaire de la vessie. L'influence des centres corticaux sur les centres réflexes, auparavant acheminé par des influx nerveux traversant la colonne vertébrale, est réduite ou inexistante. Les influx facilitateurs provenant des neurones sensitifs dans la paroi vésicale et qui provoquent la contraction du détrusor, ne peuvent être contre balancés par

des influx inhibiteurs supra-médullaires (corticaux). Le réflexe provoquant la contraction du détrusor se produit pour des volumes plus faibles (vessie hyperactive), augmentant la pression vésicale pendant le remplissage et réduisant significativement la capacité vésicale à long terme ^[49]. Également, le sphincter strié n'est plus sous contrôle volontaire ce qui élimine la continence active du processus de la miction.

Des effets indirects d'un traumatisme peuvent aussi être observés. La pression vésicale plus élevée pendant de plus longues durées, causée par l'absence des influx inhibiteurs, provoque un reflux d'urine de la vessie vers les reins par les uretères. Ce reflux urétéral et la rétention chronique entraînent des troubles rénaux et éventuellement la défaillance rénale si aucune action n'est prise. L'incontinence est également une des complications remarquées chez les patients avec une lésion à la colonne vertébrale. Aussi, le cathétérisme intermittent, solution couramment employée pour la vidange de la vessie chez les individus privés du contrôle vésical consistant à insérer un cathéter par l'urètre jusque dans la vessie, conjugué à la stase urinaire favorisent les infections urinaires à répétition. L'ensemble des effets d'un traumatisme conduit souvent à la défaillance des fonctions rénales, une cause important de mortalité chez les paraplégiques ^[44]. Elle résulte principalement de la rétention chronique et de l'infection urinaire.

1.6 La stimulation neuromusculaire

Plusieurs options sont possibles pour vidanger la vessie et réduire les effets d'un traumatisme. La technique la plus utilisée, le cathétérisme intermittent, consiste à insérer un cathéter par l'urètre jusqu'à ce que l'extrémité pénètre dans la vessie pour faciliter l'écoulement d'urine. L'utilisation quotidienne de cette technique nécessite le plein contrôle des membres supérieurs, ce qui n'est pas le cas des quadraplégiques, et favorise les infections urinaires.

La stimulation neuromusculaire est assurément l'avenue la plus prometteuse pour la récupération des fonctions urinaires. Elle consiste à exciter les tissus musculaires toujours fonctionnels entourant la vessie par l'intermédiaire des nerfs moteurs du système vésicale à l'aide de courants électriques pour provoquer la contraction du détrusor et induire la miction. Il y a présentement quatre sites connus pour provoquer la miction à l'aide de la stimulation électrique: le muscle de la vessie (détrusor), les nerfs pelviens, les racines sacrées et la colonne vertébrale. Chacun de ces sites a fait l'objet de recherches approfondies et présente plusieurs avantages et inconvénients.

1.6.1 Les sites de stimulation

Le détrusor fut le premier site de stimulation étudié [7]. Lorsqu'un stimulus électrique est appliqué par le biais d'électrodes cousues dans la paroi vésicale, le détrusor se contracte et provoque la miction. Cette méthode de stimulation très effi-

cace montre une faible résistance urétrale ^[47]. Cependant, les courants d'excitation nécessaires pour obtenir une contraction uniforme du muscle vésical doivent être élevés, ce qui, à long terme cause une désensibilisation des tissus autour des électrodes. Également, les mouvements du muscle vésical pendant le remplissage et la stimulation provoquent le bris prématuré des électrodes ^[32] ^[46].

La stimulation des nerfs pelviens gauche et droit entraîne aussi la miction. Toutefois, on peut observer une plus grande résistance urétrale que lors d'une stimulation du détrusor. Cette augmentation de résistance est due à la stimulation des neurones sensitifs afférents qui provoque une contraction du sphincter par le biais du centre réflexe sacré ^[26]. La vidange complète de la vessie est seulement obtenue après la section des nerfs honteux. Aussi, la chirurgie pour exposer les nerfs pelviens est plus difficile que les chirurgies pour les autres sites de stimulation.

La moelle épinière est le troisième site de stimulation pouvant provoquer l'évacuation de l'urine de la vessie. Des électrodes au bout non-isolé sont introduites sous la surface de la colonne vertébrale pour obtenir la stimulation de la vessie et le relâchement du sphincter produisant ainsi une miction à basse pression ^[16] ^[21] ^[22] ^[27]. Les résultats sont toutefois sensibles à la position des électrodes, ce qui réduit l'efficacité de ce site de stimulation. L'insertion d'électrodes dans la moelle épinière augmente également le risque d'infection chez les patients ^[32] ^[46].

Le site de stimulation le plus prometteur pour la récupération des fonctions vésicales reste sans doute les racines sacrées ^[39]. La stimulation électrique des nerfs

sacrés engendre la contraction de la vessie mais, comme dans le cas de l'excitation des nerfs pelviens, elle provoque également la contraction du sphincter. Les fibres somatiques qui innervent le sphincter urétral sont plus sensibles à la stimulation électrique que les fibres autonomes qui rejoignent la vessie. Ainsi, un courant assez élevé pour provoquer la contraction du détrusor activera inévitablement le sphincter réduisant ou annulant la miction [5]. Ce site de stimulation montre toutefois les résultats les plus encourageants.

1.6.2 Les techniques de stimulation des racines sacrées

Il y a présentement quatre techniques de stimulation des racines sacrées (Tableau 1.1) destinées à surmonter la dyssynergie entre le détrusor et le sphincter et provoquer la miction: la vidange poststimulus, le blocage anodique, la fatigue du sphincter et la stimulation sélective par blocage haute-fréquence [39].

1.6.2.1 La vidange poststimulus

La vidange poststimulus est une technique basée sur les temps de relaxation du détrusor et du sphincter [11] [48]. La musculature striée du sphincter externe se relâchant plus rapidement que la musculature lisse du détrusor, une stimulation des racines sacrées en utilisant un patron typique consistant en 3-6 secondes de stimulation en alternance avec 6-9 secondes de pause, provoque une vidange saccadée de la vessie.

Cette technique de stimulation est présentement la plus utilisée. Un stimulateur neuromusculaire conçu par Brindley, abordé au prochain chapitre, a été implanté dans plus de 1000 patients et donne de bons résultats cliniques.

1.6.2.2 Le blocage anodique

Le blocage anodique est une autre technique utilisée pour pallier la contraction simultanée du détrusor et du sphincter. Cette technique vise l'activation sélective des fibres parasympathiques en combinant la stimulation de toutes les fibres à un blocage anodique sélectif [32]. Comme le courant nécessaire au blocage des fibres somatiques est inférieur à celui nécessaire pour arrêter les potentiels d'action parcourant les fibres parasympathiques, l'hyperpolarisation partielle de la membrane nerveuse à l'anode de l'électrode tripolaire bloque la propagation des potentiels d'action vers le sphincter strié mais laisse passer ceux destinés au détrusor. Cette technique de stimulation prévient la dyssynergie vésico-sphinctérienne et permet la miction. La validation en phase aiguë sur des animaux montre que le blocage anodique peut réduire de plus de 80% la pression induite au niveau de l'urètre. Rijkhoff soutient également que cette technique peut être appliquée chez l'humain.

1.6.2.3 La fatigue du sphincter

La résistance urétrale lors de la miction peut être réduite par une stimulation haute-fréquence des nerfs honteux pour induire la fatigue du sphincter externe [28]

tout en maintenant une stimulation basse-fréquence au niveau des racines sacrées. Les fibres musculaires striées constituant le sphincter externe sont moins résistantes à la fatigue que la musculature lisse du détrusor. Après quelques secondes de stimulation, la pression urétrale exercée par le sphincter se relâche, permettant ainsi la miction. Des études en phase aiguë et chronique sur des chiens ont montrées l'efficacité de cette méthode en donnant des résultats comparables à un groupe ayant subi une neurectomie des nerfs honteux. La chirurgie pour atteindre les nerfs honteux est toutefois extensive.

1.6.2.4 La stimulation sélective

La dernière technique de stimulation, faisant l'objet des travaux rapportés dans ce mémoire, consiste à exciter les racines sacrées, innervant à la fois le détrusor et le sphincter, à l'aide d'un signal composé de deux trains d'impulsions bipolaires superposés ^[41] [9]. Un premier train d'impulsions basse-fréquence et d'amplitude élevée provoque la contraction du détrusor tandis que le second train d'impulsions haute-fréquence et d'amplitude réduite inhibe la contraction du sphincter. Les fibres nerveuses somatiques, contrôlant le sphincter, ont un seuil d'activation plus bas que les fibres parasympathiques innervant le détrusor. En choisissant une amplitude inférieure au seuil des fibres parasympathiques mais excitant les fibres somatiques, il est possible de provoquer, d'une part le relâchement du sphincter externe, et d'autre part, la contraction du détrusor afin d'entraîner la miction.

Tableau 1.1: Résumé des techniques de stimulation des racines sacrées dédiées à la vidange de la vessie.

Technique	Avantages & Inconvénients	État
Vidange poststimulus	(+) 1 site d'implantation; (-) Pression intravésicale élevée; (-) Miction saccadée	Utilisation clinique
Blocage anodique	(+) 1 site d'implantation	Essais en phase aiguë
Fatigue du sphincter	(+) Faible dyssynergie; (-) 2 sites d'implantation	Essais en phase aiguë; Technique obsolète
Stimulation sélective	(+) Faible dyssynergie; (+) 1 site d'implantation	Essais en phase chronique

1.7 Conclusion

Ce premier chapitre couvre les notions essentielles au design d'un microstimulateur implantable pour la récupération des fonctions vésicales. En apportant une connaissance approfondie des différents mécanismes oeuvrant lors de la miction normale et des problèmes amenés par une lésion à la colonne vertébrale, elle permet de mieux comprendre et d'analyser les résultats expérimentaux chez l'animal.

CHAPITRE 2

LE SYSTÈME DE STIMULATION NEUROMUSCULAIRE

2.1 Introduction

Parallèlement à l'évolution des techniques de stimulation présentées au chapitre précédant, les systèmes de stimulation neuromusculaire dédiés à la récupération des fonctions vésicales ont connu un développement continu. Le présent chapitre est consacré à la revue de systèmes disponibles commercialement ainsi que des neurostimulateurs et contrôleurs proposés par l'équipe PolySTIM^[18]. Un bref aperçu des différentes électrodes est aussi donné.

2.2 Les systèmes commerciaux

Le «Finetech-Brindley Bladder System» est un des produits disponibles pour la récupération des fonctions urinaires ^[10]. Ce système comprend un stimulateur implantable, composé de trois récepteurs identiques, et un contrôleur externe. Il génère une excitation basse fréquence en alternance avec des périodes de repos pour provoquer une vidange poststimulus. L'énergie nécessaire à la stimulation est transmise par le contrôleur externe par lien RF lors de l'activation de l'implant.

La thérapie InterStim[®] proposée par Medtronic, Inc. utilise également un neurostimulateur pour provoquer la vidange de la vessie. Cet implant, alimenté par

piles, offre une gamme de fréquences de 2,1 à 130 Hz avec une largeur d'impulsions allant de 60 à 450 μ s et une amplitude variant entre 0 et 10,5 volts. Il permet aussi d'utiliser une électrode bipolaire ou quadripolaire. La programmation et le test du neurostimulateur implantable sont réalisés à l'aide d'un programmeur destiné aux médecins tandis que l'activation quotidienne se fait à l'aide d'un aimant et est utilisé par le patient. Cet implant est aussi doté d'un système de télémessure pour confirmer la réception des paramètres [1].

Enfin, un système appelé «Urogenital Stimulator» est offert par Avery Laboratories, Inc. (Le groupe Dobelle). Il comprend lui aussi un contrôleur externe et un stimulateur implantable [23] mais peu de détails sont toutefois disponibles sur ce système.

2.3 Aperçu sur l'évolution de stimulateurs offerts par PolySTIM

L'équipe PolySTIM a mis en oeuvre plusieurs stimulateurs au cours de la dernière décennie. Un bref aperçu des différentes parties implantables des systèmes de stimulation neuromusculaire est donné dans cette section. Ces systèmes sont présentés dans un ordre chronologique.

Le point de départ des neurostimulateurs dédiés à la récupération des fonctions vésicales de l'équipe PolySTIM est un système introduit par Sawan dans le cadre de sa thèse de doctorat^[35]. Il s'agit d'un stimulateur implantable à 8 canaux monopolaires (ou 4 canaux bipolaires) construit à partir d'un microprocesseur dédié

à instructions de 24 bits et réalisé sur prédiffusé de $4,0 \mu\text{m}$ (figure 2.1). Il offre une large gamme de fréquences et une amplitude maximale de 3,0 mA. L'énergie de l'implant est fournie par le contrôleur externe. Le patron de stimulation reste toutefois limité car l'amplitude pour chacun des canaux est envoyée en temps réel [38], [37], [36], [39].

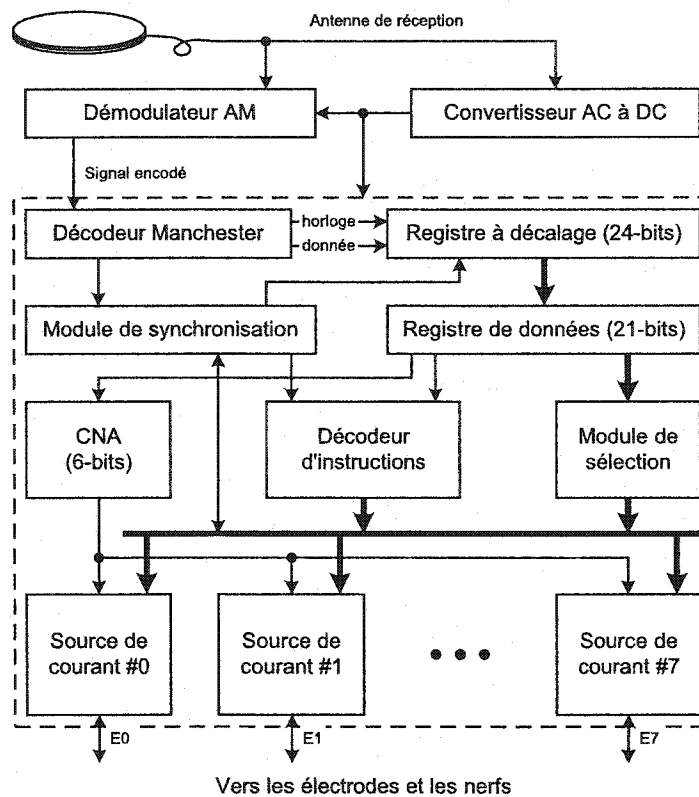


Figure 2.1: Diagramme simplifié du neurostimulateur introduit par Sawan.

Un neurostimulateur conservant les principales caractéristiques du produit précédant et offrant, entre autres, une nouvelle fonction permettant la modulation en fréquence de stimuli a été réalisé par Arabi [4]. Ce circuit, conçu sur CMOS standard de $1,2 \mu\text{m}$ et présenté à la figure 2.2, permet de détecter et de corriger

une erreur lors de la transmission d'un paramètre de stimulation. Les stimuli pour chacun des canaux (8 monopolaires ou 4 bipolaires) sont générés localement, à partir des paramètres reçus, par quatre instances d'un circuit numérique dédié à la génération des stimuli. Ce stimulateur, utilisé pour la vidange de la vessie par fatigue du sphincter, n'offre toutefois pas la flexibilité nécessaire à la stimulation sélective.

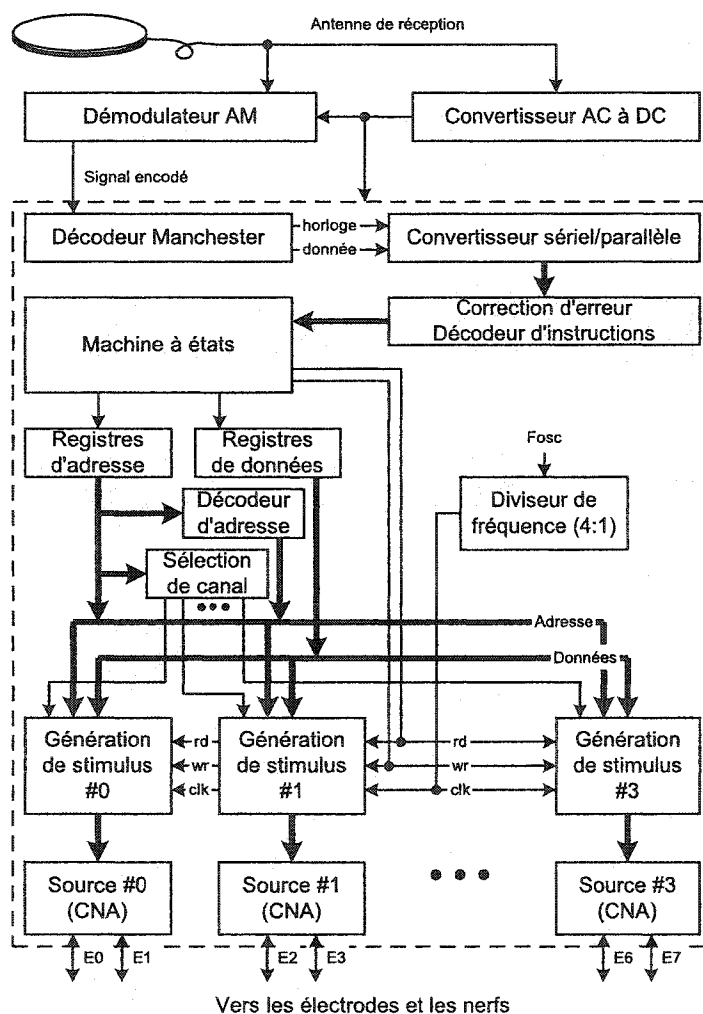


Figure 2.2: Diagramme simplifié du neurostimulateur introduit par Arabi.

Pour combler les besoins nécessaires aux nouvelles techniques de stimulation, un microstimulateur neural sélectif (MSN) fut complété par Robin [33], [34]. Cet implant à un seul canal, réalisé à l'aide d'un FPGA et de composants discrets, produit un signal bipolaire composé de deux fréquences superposées (figure 2.3). Il offre un courant maximal de stimulation de 2,5 mA, une gamme de fréquences comprise entre 10 Hz et 1 kHz et une largeur d'impulsion allant jusqu'à 300 μ s. L'énergie nécessaire à la stimulation est fournie par un contrôleur externe dédié.

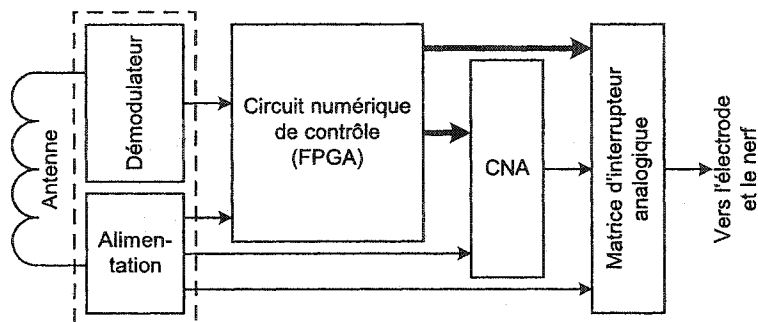


Figure 2.3: Schéma bloc du neurostimulateur introduit par Robin.

Les travaux qui ont suivi, menés par Schneider, ont donné lieu à deux nouvelles versions de ces stimulateurs implantables. Le premier, un neurostimulateur à lien bidirectionnel (NLB) introduit un outil de diagnostique [40]. Il permet d'évaluer la composante résistive de l'interface nerf-électrode *in vivo* et d'envoyer les résultats de la mesure vers le contrôleur à l'aide du lien RF bidirectionnel. La télémessure utilise des courants inférieurs au seuil d'activation des fibres sympathiques et parasympathiques pour éviter la contraction de la vessie lors de la vérification de l'interface nerf-électrode. Les gammes des paramètres de stimulation restent sensiblement les

mêmes que pour le MSN soit un courant maximal de 2,0 mA, une sortie bipolaire multifréquence variant entre 18 Hz et 2 kHz et une largeur d'impulsion maximal de 210 μ s. Cet implant, réalisé avec un FPGA et d'autres composants discrets, récupère l'énergie nécessaire à son fonctionnement par le lien inductif.

Le second, un neurostimulateur sélectif et permanent (NSP), également basé sur le MSN, supporte deux modes de stimulation, soit la stimulation sélective pour la vidange de la vessie et une stimulation permanente pour aider à la rétention de l'urine [40]. Le NSP, réalisé à l'aide d'un FPGA, d'un microcontrôleur de la série PIC et d'autres composants discrets, reçoit l'énergie nécessaire à la stimulation sélective du contrôleur externe mais utilise une pile pour la stimulation permanente.

Le Tableau 2.1 résume les principales caractéristiques des stimulateurs électriques implantables développés par les membres de l'équipe PolySTIM.

Tableau 2.1: Résumé des caractéristiques des neurostimulateurs implantables.

Stimulateur	Technologie	# canaux	Fréquence	Amplitude
Sawan	4,0 μ m	8/4 ^a	1-500 Hz	3,0 mA
Arabi	1,2 μ m	8/4 ^a	1-2000 Hz	-
Robin	0,8 μ m	1 multifréquence	10-1000 Hz	2,5 mA
Schneider	discret	1 multifréquence	18-2000 Hz	2,0 mA

^a8 canaux monopolaires ou 4 canaux bipolaires

2.4 Les contrôleurs externes

Bien que le mode de communication entre les stimulateurs implantables et leur contrôleur externe soit resté le même depuis les premières versions (i.e. modulation

ASK avec encodage Manchester), le format des trames, les gammes de chacun des paramètres et les options disponibles ont changé. Cette section présente brièvement les différents contrôleurs externes réalisés par l'équipe PolySTIM pour chacun des neurostimulateurs implantables. Ils sont présentés dans le même ordre que les implants de la section précédente, soit chronologiquement.

Le premier contrôleur externe destiné aux implants utilise un ordinateur personnel (PC) comme interface primaire. Un autre contrôleur, complété dans la même période et appelé processeur externe pour une prothèse urinaire (PEPU), inclus un microcontrôleur MC68HC805 de Motorola, un clavier de 20 touches, un écran LCD de 16 caractères ainsi qu'un système de régulation de l'alimentation pour produire les tensions nécessaires à son fonctionnement à partir de piles^[37],^[39]. Ce contrôleur offre le maximum de flexibilité pour la modification des paramètres de stimulation pour chacun des 8 canaux.

Un nouveau contrôleur externe, utilisant également un PC comme interface usager, fut développé pour le MSN^[34]. Son architecture simplifiée, basée sur un FPGA, offre une flexibilité de programmation maximale grâce à l'utilisation du PC. Le contrôleur reçoit la trame de 64 bits du port parallèle du PC nécessaire à sa programmation et envoie les données à l'implant à travers le lien RF inductif.

Une version portative du contrôleur précédent, également destiné au MSN, permet de choisir parmi six ensembles de paramètres préprogrammés dans une mémoire morte de type EPROM^[34]. Ce contrôleur, basé sur une architecture utilisant un

FPGA, incorpore un écran LCD de 2 lignes et un clavier de 4 touches.

Conjointement avec le développement du NLB et du NSP, un contrôleur de test flexible utilisant un PC comme interface usager fut réalisé. Ce contrôleur, ayant la même architecture que le système développé pour le MSN, permet d'envoyer une trame de longueur variable pouvant contenir jusqu'à 88 bits ^[40].

Le besoin de systèmes moins encombrant pour une utilisation quotidienne a motivé le design d'un contrôleur portatif simplifié (CPS). Cette nouvelle architecture utilise un microcontrôleur PIC, quelques boutons et un écran LCD de 2 lignes ^[40]. La même architecture a donné lieu à une deuxième version du contrôleur avec une nouvelle programmation. Cette fois, un écran LCD de 4 lignes et des boutons de menu contextuel offrent une flexibilité accrue pour le choix des paramètres de stimulation. Ces systèmes sont également dédiés au NLB et au NSP.

Tableau 2.2: Résumé des caractéristiques des contrôleurs externes.

Stimulateur	Interface	Communication	Modulation
PEPU	μ contrôleur	Simplex	ASK
Arabi	PC μ contrôleur	Duplex simultané RF - optique	ASK
Robin	FPGA	Simplex	ASK
Schneider	μ contrôleur PIC	Duplex simultané RF	ASK

2.5 Les électrodes

Le système de stimulation neuromusculaire nécessite également une électrode pour acheminer les courants de stimulation entre l'implant et le nerf. Un bref

aperçu des types d'électrodes disponibles est donné plus bas. L'exercice ne vise que les électrodes à gaine (i.e. dont la forme entoure le nerf). Pour de plus amples détails, vous pouvez consulter les travaux de Crampon [17].

Un premier type d'électrodes à gaine cylindrique utilise des fils de sutures noués pour assurer le positionnement de l'électrode autour du nerf [25]. C'est le type d'électrodes le plus facile à réaliser. D'autres types d'électrodes utilisent les alliages à mémoire de forme (figure 2.5) pour retrouver et conserver leur forme cylindrique après leur installation. Enfin, on peut également trouver des électrodes en spirale, à gaines cylindriques concentriques ou hélicoïdales.

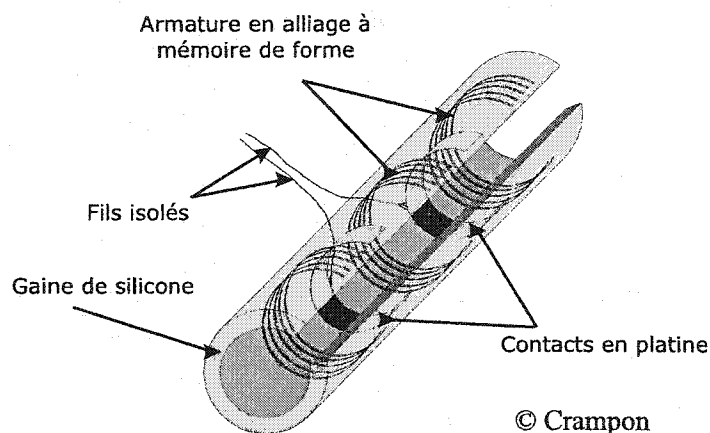


Figure 2.4: Électrode à gaine cylindrique avec armature en alliage à mémoire de forme réalisée par Crampon.

2.6 Conclusion

La revue des stimulateurs neuromusculaires existant permet de passer au coeur du mémoire, soit l'article «Implantable Selective Stimulator to Improve Bladder

Voiding: Design and Chronic Experiments in Dogs». Les données présentées dans cet article proviennent d'expérimentations en phase chronique réalisées sur 8 chiens à l'aide d'un microstimulateur implantable réalisé à partir de composants discrets.

CHAPITRE 3

SIMULATEUR IMPLANTABLE ET VALIDATION DE LA TECHNIQUE DE STIMULATION SÉLECTIVE EN PHASE CHRONIQUE

3.1 Introduction à l'article publié dans «IEEE Transactions on Rehabilitation Engineering»

Le présent chapitre constitue le coeur de ce mémoire. Il s'agit d'un article publié dans la revue «IEEE Transactions on Rehabilitation Engineering» ^[9] qui passe en revue les sites de stimulation dédié à la récupération des fonctions vésicales ainsi que les techniques utilisées pour contourner les problèmes reliés à la dyssynergie de détrusor et du sphincter urétrale externe lors de la stimulation des racines sacrées. Il donne également une appréciation des résultats obtenus tant au niveau technique, avec la caractérisation des circuits fabriqués par l'équipe PolySTIM, qu'au niveau médical, en présentant les résultats d'essais en phase chronique sur huit chiens bâtards.

3.2 Article «Implantable Selective Stimulator to Improve Bladder Voiding: Design and Chronic Experiments in Dogs»

3.2.1 Abstract

Among the treatments to enhance the bladder voiding, the sacral roots neurostimulation is one of the most promising techniques. The electrostimulation of sacral nerves provokes a simultaneous contraction of the detrusor muscle as well as the external urethral sphincter (EUS). A new simplified-architecture implantable stimulator with its wireless controller have been designed to investigate high-frequency inhibition stimulation strategies. This innovative technique based on high-frequency inhibition reduces sphincter activity during stimulation. Low-frequency current pulses also applied to the sacral roots induce contraction of the detrusor muscle resulting in low pressure voiding. Chronic experiments were carried out on ten male mongrel paraplegic dogs. One cuff electrode was implanted along with each stimulator for eight months. The animals were stimulated twice a day using the prototypes of our implantable selective stimulator while voided and residual urine volume were measured during the procedure. These experiments revealed that the proposed stimulation strategy enhances bladder voiding by more than 50% in comparison with low-frequency only stimulation. The residual urine volume was reduced to an average of 9% and low pressure micturition was achieved as shown by weekly cystourethrograms.

Keywords: Selective electrical stimulation, urinary bladder, implantable stimulator, chronic experiments, sacral roots.

3.2.2 Introduction

Several functional electrical stimulation (FES) techniques have been introduced to promote bladder voiding and prevent incontinence [8] [45] [50]. Clinical experiments as well as animal validation have been carried out to address the bladder voiding function. Four currently known stimulation sites (Fig. 3.1) have been investigated: the bladder wall (detrusor muscle), pelvic nerves, sacral roots and the spinal cord. Each one of these stimulation techniques presents its advantages and drawbacks [32] [39] [46] [47].

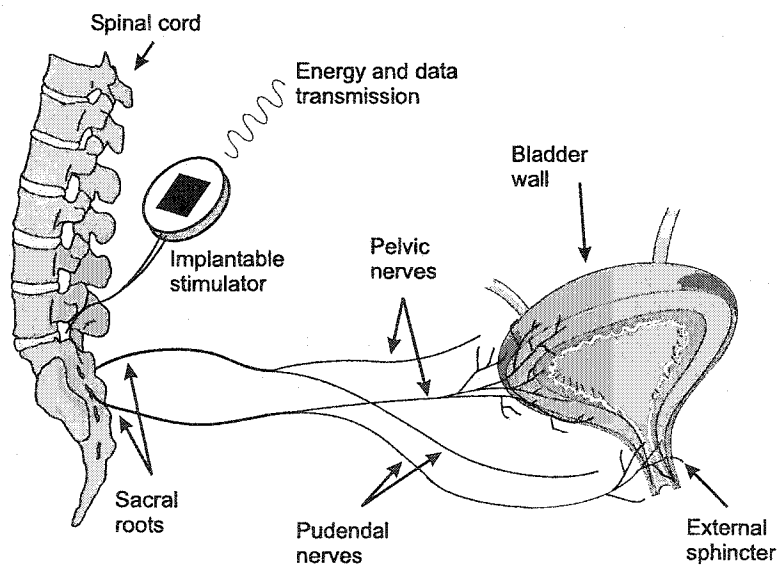


Figure 3.1: Possible stimulation sites and implantable stimulator position for high-frequency inhibition.

Bladder wall stimulation was the first investigated method to induce micturi-

tion [7]. Electrodes were sewn in the detrusor muscle and electrical stimulation was applied to induce detrusor muscle contraction but the movement of the detrusor muscle during bladder filling and emptying causes frequent electrode breakage while stimulation using high current amplitude increases impedance around the electrode and may cause damages [32] [46].

Electrostimulation by the right and left pelvic nerves could induce bladder voiding but many problems were reported. The external urethral sphincter was activated during excitation and resulted in impaired micturition. Complete emptying of the bladder was only obtained after section of the pudendal nerves [26]. Also, the surgery needed to expose the pelvic nerve is difficult compared to the surgery for other stimulation methods.

The third stimulation site to achieve micturition is the spinal cord. Using electrodes with exposed tips to reach beneath the spinal cord dorsal surface, selective activation of the detrusor muscle and external urethral sphincter relaxation can be obtained, producing low pressure bladder voiding [16] [21] [22] [27]. However, results were affected by electrode positioning and there was a high infection rate due to electrode insertion in the spinal cord [32] [46].

Regarding the fourth stimulation site, it is well established that sacral roots are the most promising electrode implantation sites to achieve bladder voiding functions [32] [39] [46]. Like the stimulation of the pelvic nerves, sacral roots excitation induces sphincter contraction which impairs micturition. Somatic fibers that supply

the urethral sphincter are more sensitive to electrical stimulation than autonomic fibers innervating the detrusor muscle. This means that a current large enough to provoke detrusor muscle contraction will inevitably activate the external urethral sphincter and interfere with micturition [5]. However, this site of stimulation shows the most encouraging results.

3.2.2.1 Neurostimulation Techniques of the Bladder

Currently, there are four main electrostimulation techniques used to overcome this dyssynergia and provoke micturition: poststimulus voiding in conjunction with dorsal rhizotomy, anodal block, sphincteric fatigue and selective stimulation by high-frequency inhibition [39].

1) *Poststimulus Voiding*: The poststimulus voiding technique is based on the fact that the striated sphincter muscle relaxes more rapidly than the smooth detrusor muscle. The sacral roots stimulation pattern consists of intermittent pulse trains (typically 3-6 seconds stimulation and 6-9 seconds pause) to allow bladder evacuation. An implantable stimulator designed by Brindley is presently one of the most clinically used systems to restore partial bladder functions. It has been used in over 700 patients with good clinical results [11] [48]. However, poststimulus voiding necessitates the transection of neural pathways (posterior rhizotomy) to impede reflex detrusor-sphincter dyssynergia (DSD) and allow voiding during the pause following stimulation as reported by Brindley [11]. The dorsal rhizotomy

also improves bladder compliance and capacity and prevents reflex incontinence. It does not affect stimulation-induced DSD but will abolish any reflex erection if previously present.

2) *Anodal Block*: This method of neurostimulation consists of preventing the propagation of the nerve action potential toward the external urethral sphincter using an anodal block technique. This technique uses hyperpolarization of the nerve membrane between the excitation application point and the external urethral sphincter [2] [20]. Acute animal experiments showed that the anodal blocking technique could decrease by more than 80% the stimulus-induced intraurethral pressure. Rijkhoff *et al.* also state that the technique can be applied to humans [32].

3) *Sphincteric Fatigue*: Urethral resistance can be reduced by stimulating pudendal nerves with high-frequency signals to induce sphincteric fatigue, while low-frequency current pulses are applied at the sacral root level, allowing micturition [28] [29] [39]. This technique, based on the fact that high-frequency stimulation rapidly provokes fatigue of the striated external urethral sphincter, was performed in both acute and chronic experiments on dogs. It achieved proper bladder emptying without performing neurectomy but necessitated surgery to reach the pudendal nerves. Results from the chronic experiments on dogs with the sphincteric fatigue stimulation strategy at the pudendal level were comparable to the results obtained on a control group with pudendal neurectomy [29].

4) *Selective Stimulation by High-Frequency Inhibition*: Recently, selective de-

trusor muscle activation has been obtained by performing stimulation of the sacral roots with a signal composed of two distinctive trains of bipolar-current pulses [41]. A high-amplitude, low-frequency train provokes detrusor muscle contraction while a low-amplitude, high-frequency train inhibits the external urethral sphincter contraction to allow micturition. This stimulation method, which is the subject of this article, allows bladder evacuation with a low-pressure voiding and low residual urine.

Previous attempts in primates to fatigue the sphincter with low-frequency (20 Hz) stimulation was not successful with poor voiding. The fatigued striated musculature sufficed for continence against maximal bladder contraction. However, the urethral pressure obtained with higher frequencies was significantly reduced resulting in better voiding [10].

3.2.2.2 Available Stimulators

Regarding the technical aspect, many devices are now available for neuromuscular stimulation but they lack important features:

1. a wide range of programmable parameters,
2. a high efficiency in energy transfer and data transmission,
3. a user-friendly interface,
4. high-frequency stimulus generation and,

5. waveform flexibility (monophasic, biphasic, anodic, etc.)

The two most frequently used commercially available stimulators for bladder control are :

- Medtronic: an implant programmable by two devices (physician full range programming and patient amplitude programmer). It also contains a telemetry circuit to confirm data validation and is limited to a maximum frequency of 130 Hz ^[1].
- NeuroControl Corporation: an implant designed by Brindley and composed of several identical receivers with their inductive couplings. Its maximum frequency is 300 Hz ^[10].

Several implantable multichannel stimulators and their external controllers were proposed by members of our team to fill the gaps and perform sacral roots and pudendal or sacral nerve stimulation ^{[37] [36] [39]}. These systems, validated in previous experiments, allowed to generate a wide range of stimuli through miniaturized implants ^{[4] [37]}. More recently, our team proposed a new stimulator based on a simplified architecture which is dedicated to selective stimulation applications. This device has been used in acute experiments and is validated in the current study. It can generate stimuli composed of two different bipolar-current trains of pulses with a high-frequency capability up to 1 kHz.

This paper details the combined sacral root stimulation of the bladder by low-frequency with high-frequency sphincter inhibition. The employed implantable stimulator is summarized in Section 3.2.3, the stimulation protocol and methods are described in Section 3.2.4, results of chronic experiments on 10 male mongrel dogs are reported in Section 3.2.5 and discussed in Section 3.2.6.

3.2.3 Description of the Stimulation System

Our stimulation system is composed of two main parts ^[34]:

- The external controller providing digital information, operating clock and energy to the implant through the skin;
- The implantable stimulator generating the current pulses through a bipolar cuff electrode which is wrapped around the sacral nerve.

Transcutaneous link through an optimized inductive coupling technique ^[19] allows the implantable part to generate a stimulus composed of two waveform trains of bipolar-current pulses defined by their amplitudes, frequencies and pulse widths. The generated waveforms are mixed together in order to produce well balanced stimuli in term of charge quantities as well as the number of complete cycles for each train of pulses (Fig. 3.2).

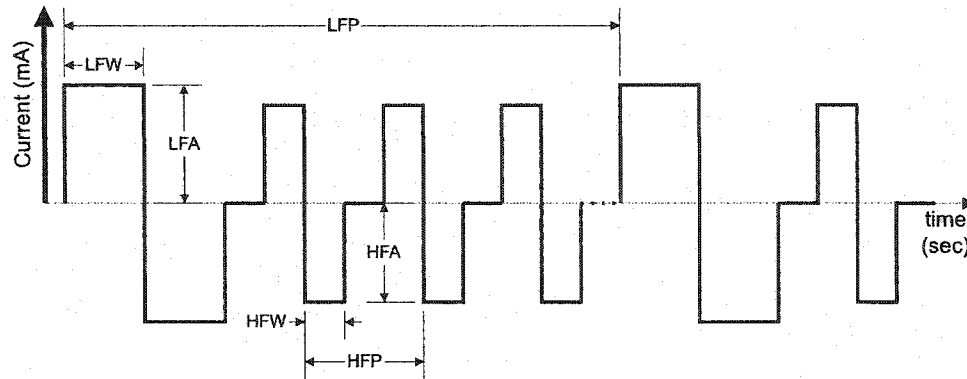


Figure 3.2: Typical selective stimulation waveform intended to block the sphincter activation during bladder contraction (waveforms are not to scale; meaning of abbreviations are given in Table 3.1).

3.2.3.1 The External Controller

Two complementary models of the external controller have been realized. The first one, based on a personal computer (PC), allows generation of an unlimited sets of parameters and was developed to facilitate prototyping and testing of stimulators. This PC interfaced controller offers more flexibility to adjust the parameters for each animal. Through a dedicated software, each stimulation parameter can be quickly viewed and changed. The selected set of parameters can be saved and retrieved from disk or even exported to a binary file for further use with the hand-held model [34].

The second one, a memory based hand-held unit, is a dedicated mobile unit. The hand-held unit, based on a field-programmable gate array (FPGA) and an EPROM (Fig. 3.3), includes eight different stimulation sets which can be easily reprogrammed. The parameter values and the text to guide the user to choose

one of these eight sets of parameters are also stored in memory while the FPGA integrates a finite state machine (FSM) and other logic blocks to control the stimulation system. The simplified architecture of the proposed controller allows rapid printed circuit board (PCB) design and easy to modify and to implement functions. The user friendly interface consists of a 2 x 16 characters dot matrix liquid crystal display (LCD) and a 4-touch keyboard. To select a set of parameters for stimulation, the user must choose within the preprogrammed sets. The values of the active parameters of each programmed set are displayed on the LCD. To run a stimulation period, the antenna of the external controller must be aligned with the antenna of the implant and a start key has to be activated.

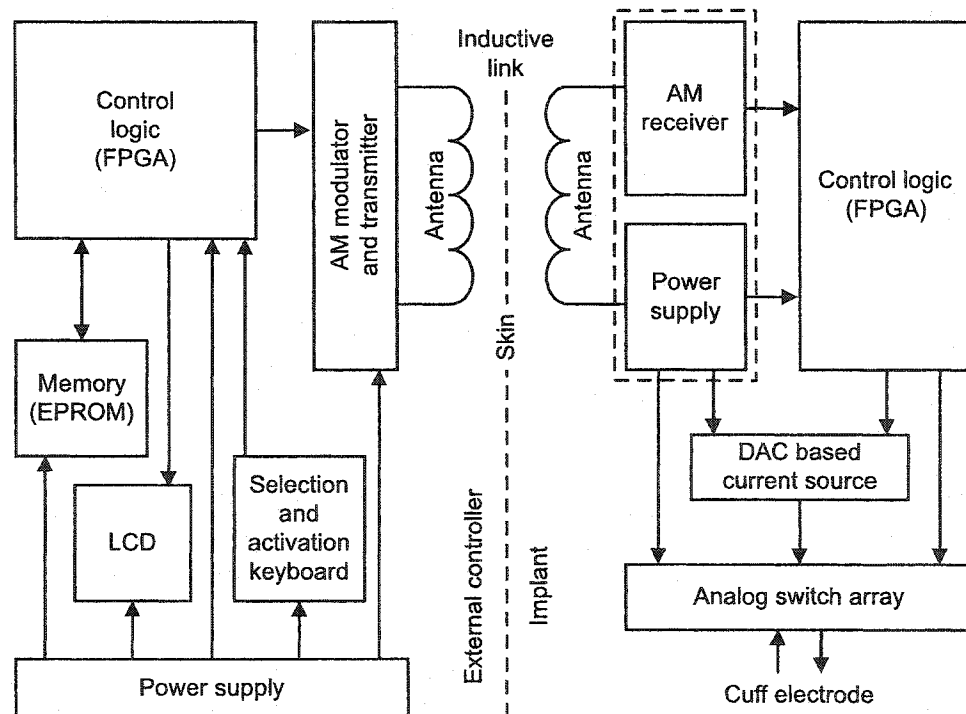


Figure 3.3: Block-diagram of the whole proposed selective stimulator including the external controller, the implant and the electromagnetic link between both parts.

For both models of the external controller, the clock and the command words form the serial data which is encoded in Manchester format ^[36]. The encoded bit stream and the needed energy to power up the implantable stimulator are sent through an inductive link (Fig. 3.3) based on the radio-frequency (RF) coupling technique. The output stage is composed of a class D amplifier and a power regulator to compensate for coupling factor variation due to displacement between emitting and receiving coils ^[39]. The link uses a 20 MHz carrier modulated in amplitude.

3.2.3.2 The Implantable Stimulator

The implantable part of the stimulation system (the implant) receives data and energy through the inductive coupling link. The received AM waveform is rectified and regulated to power up the implant while the clock signal and data are extracted from the Manchester encoded signal. The digital modules of the implant are implemented in an FPGA which is placed at the center of the 4-cm circular PCB. The remaining analog and mixed-signal (digital / analog) parts of the implant are realized using commercially available components of surface mounted technology formats. The integrated digital part of the implant detects the header of each received data and generates the control commands to be interpreted by the output stage. To transform the digital signal into a current waveform, the output stage includes a digital to analog converter (DAC), a voltage to current converter, a

current amplifier and an analog switch array (Fig. 3.4) to generate a fully balanced bipolar stimulus which is necessary to prevent polarization of the nerve by charge injection and accumulation at the electrode-tissue interface.

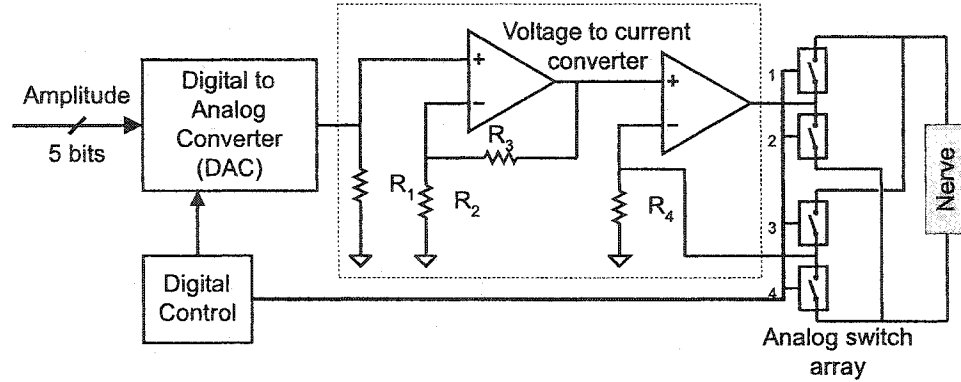


Figure 3.4: Simplified block diagram of the output stage of the implant including an analog switch array to deliver fully balanced bipolar stimuli.

3.2.3.3 The Implant to Nerve Connection

The stimulator is connected to sacral nerves through dedicated cuff electrodes. The electrode consists of two stainless-steel leads covered with polytetrafluoroethylene (TeflonTM) and soldered to 25 μm platinum foil forming the contact surface to the nerve inside the cuff of the electrode [25]. To connect the electrode to our implantable stimulator, connectors have been added in our laboratory. Like the cuff, the connectors were also covered with biocompatible silicone elastomer. Two versions of electrodes were used for chronic experiments. The first version was built around stainless-steel leads composed of 7 strands (Cooner Wire Co. AS632) while in the second version, the leads were composed of 40 strands (AS634) to increase

the electrode resistance to stress. These electrodes offered a 1 mm wide contact surface to the nerve and an inner diameter of 1.5 mm. Electrodes were 3 mm apart and the 10 mm long electrode cuff was closed with sutures.

3.2.4 Experimental Protocol

The chronic study was approved by the McGill Animal Care Ethics Committee and conducted on 10 adult male mongrel dogs at the Animal Resource Center of the same faculty. Animals were subjected to laminectomy at the level of T10 vertebra and the spinal cord was sectioned under direct vision. The procedure was carried out under general anesthesia and aseptic techniques. At the same setting, a limited sacral laminectomy was performed and the sacral roots were identified. The extradural ventral sacral nerves supplying the urinary bladder and external sphincter were hooked and stimulated with an external pulse generator (SD9 Stimulator, Grass Medical Instruments). The intravesical pressure was measured through a triway 7 F catheter connected to a portable urodynamic analyzer (UDS-120, Laborie Medical Tech. Inc.) and a computer. After identification of the proper S2 sacral root (left or right), one cuff electrode was wrapped around the selected nerve and the stimulator was implanted subcutaneously in the flank of the animal. After the surgery, animals were placed in a sling for 3 to 4 days to prevent pressure sores until the animals adapted to their new state. The next day following surgery, twice daily stimulations with measurement of voided and residual urine were carried out using

different sets of parameters. Weekly cystourethrogram and monthly intravenous urography were also performed.

The stimulation protocol was divided into three main steps:

3.2.4.1 Low-frequency Only Stimulation (Standard Stimulation)

During the first month following implantation, the animals were stimulated with low-frequency only current pulses (3-5 periods of 10 seconds stimulation with a minute of rest period in between). Catheterization was used to evacuate remaining urine. After the shock phase, the animals usually developed bladder hyperreflexic contractions with reduction of the bladder capacity.

3.2.4.2 Selective Stimulation

The selective stimulation parameters were selected following a cystometric evaluation of the intravesical and intraurethral pressure measurements with electromyographic (EMG) recording of the pelvic floor muscles. The bladder was filled with sterile saline solution until it leaked to evaluate bladder capacity. Afterward, half the volume of saline solution was evacuated. Different sets of selective stimulation parameters for high-frequency inhibition were tried in order to select the most suitable set of stimulation parameters for each dog. The selection of parameters was determined by maximum bladder evacuation with high intravesical pressure and low intraurethral pressure. To avoid movement, the animal was under light

sedation during cystometric evaluation. For the next six to eight months, the animals were stimulated with the corresponding set of parameters twice a day. Voided and residual urine volumes were measured. Table 3.1 shows typical employed values of waveform amplitudes, pulse widths and frequencies for both standard (low-frequency only) and selective stimulation. Weekly cystometric study to monitor intravesical and intraurethral pressures and monthly intravenous urography (IVU) to visualize both kidneys, ureters and the bladder were performed for each animal. These procedures were also carried out under light sedation. Voiding cystourethrogram (VCUG) with neurostimulation to monitor vesical neck opening and urethral filling and to rule out vesicoureteral reflux was carried out after IVU study.

Tableau 3.1: Frequently employed parameters: low-frequency only and selective stimulations.

Parameters	Symbol	Low-frequency only	Selective stimulation
Amplitude	LFA	0.9 mA	0.9 mA
Period	LFP	33.3 ms ^a	33.3 ms
Pulse width	LFW	175 μ s	175 μ s
Amplitude	HFA	—	1.1–1.3 mA
Period	HFP	—	1.67 ms ^b
Pulse width	HFW	—	60–100 μ s

^a 30 Hz.

^b 600 Hz.

3.2.4.3 Post-selective Low-frequency Only Stimulation

During the last month, daily stimulations were performed with low-frequency only current pulses to compare both selective and low-frequency only stimulations.

All morning and afternoon stimulations for all phases of the experiment consisted of 3-5 period of 10 seconds stimulation with a minute rest period in between followed by collection and measurement of voided urine and catheterization to evacuate remaining urine residue which was also measured. In addition, nerve histology was performed on each animal at the end of the experiment.

3.2.5 Results

The results reported in the following section concern the electronic devices performances as well as the animal (dogs) experiments.

3.2.5.1 Selective Stimulation System

The external controller of the proposed stimulator is either a PC interface or a hand-held unit. These two controllers met the expected specifications and have been used to command the numerous implants working over the two year period assigned for the chronic study.

Regarding the internal part of the proposed stimulation system, a total of six implants have been used and implanted through the duration of the experiments. Five implants (83%) completed the study and showed high reliability. They were still functional when retrieved from the animals and most of them have been implanted in more than one animal.

In four dogs out of ten that started the study, stainless-steel leads connecting

the implant to the platinum cuff were broken at the level of the connectors. In two of these four cases (dog 4 and dog 5), the contralateral nerve was identified and a new cuff electrode was placed successfully. The other two animals did not complete the study. To resolve the encountered problem, a second version of cuff electrodes was designed and all electrodes of that version showed good mechanical properties in terms of flexibility and durability. All eight electrodes showed the same characteristics after the chronic experiment duration.

3.2.5.2 Cystometric and Radiological Results

For the eight dogs that completed the study, the average residual volume with low-frequency only stimulation was 275 ml (70% of full bladder capacity) during the first month and 133 ml (58%) the last month. Using selective stimulation during the remaining chronic experiment period, the residual volume dropped to 27 ml. This selective stimulation technique increased the mean voided urine volume by more than 50% and reduced the mean residual urine volume to 9%. In addition, no animal showed backpressure on the ureters or kidneys as evidenced by monthly radiological investigation (IVU, VCUG). As expected with high-frequency inhibition, external urethral sphincter relaxed with selective stimulation as shown by X-ray pictures (Fig. 3.5) taken before, during and after stimulation.

Table 3.2 depicts results mentioned above by showing voided urine quantities with both low-frequency only and selective stimulations. This table is divided into

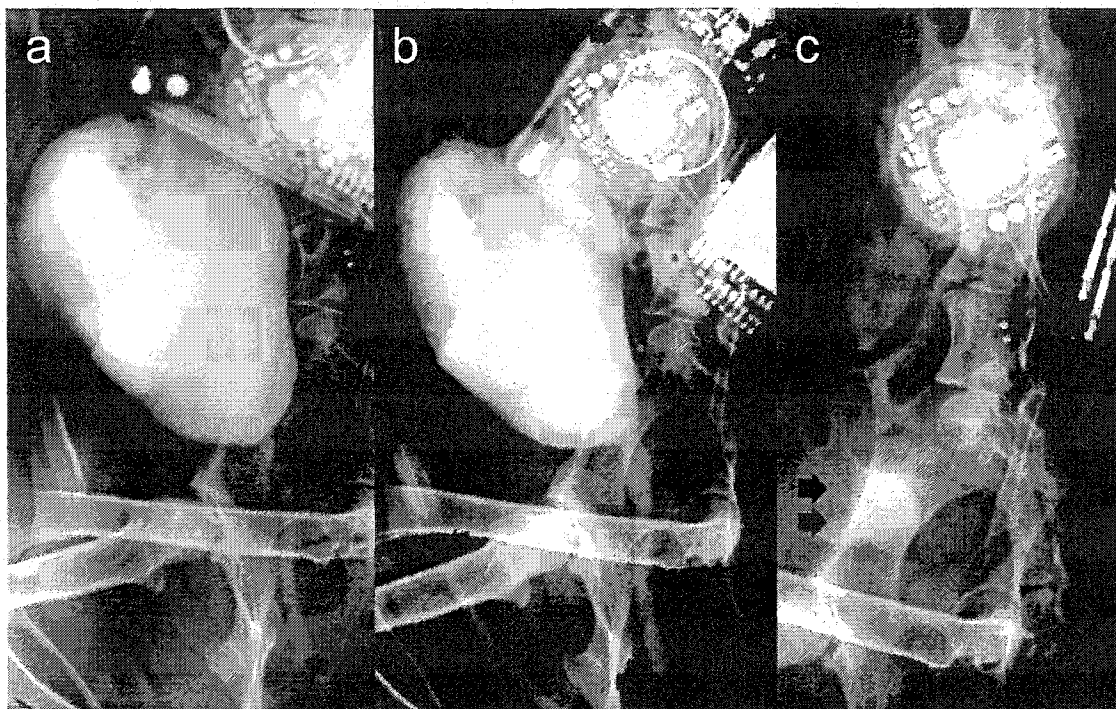


Figure 3.5: X-ray views of the bladder: a) external urethral sphincter (EUS) contracted, b) EUS relaxes with selective stimulation and, c) residual urine volume (shown by arrows).

three parts from top to bottom corresponding to the three steps of the experimental protocol: low-frequency only stimulation during the shock stage (first month), selective stimulation combining both detrusor muscle contraction and inhibition of the external urethral sphincter contraction (6-8 months) and post-selective low-frequency only stimulation (last month). In each section, the average voided and residual urine volumes along with standard deviations are shown. Figures 3.6 and 3.7 are more explicit graphics comparing low-frequency stimulation, selective stimulation and post-selective stimulation. Figure 3.6 depicts individual results from each dog as well as the average of all dogs with every stimulation type while

Tableau 3.2: Voided and residual urine quantities in eight chronic dogs.

Low-frequency stimulation	Voided urine	Residual urine	Bladder capacity
Average ^a	98 ^b	275	373
Average (%)	30%	70%	100%
Std. dev.	52	99	84
Selective stimulation	Voided urine	Residual urine	Bladder capacity
Average	247	27	274
Average (%)	91%	9%	100%
Std. dev.	49	17	52
Post-selective stimulation	Voided urine	Residual urine	Bladder capacity
Average	92	133	226
Average (%)	42%	58%	100%
Std. dev.	24	24	27

^a Average of all dogs.^b Volumes are in milliliters.

figure 3.7 shows typical intravesical and intraurethral pressures evolution in time for low-frequency only and selective stimulations. During low-frequency only stimulation, the high intraurethral pressure inhibits micturition. However, with selective stimulation, intraurethral pressure decreases in accordance with the EUS relaxation shown in Figure 3.5, while intravesical pressure remains almost constant. In addition to bladder evacuation, lower limb muscle contraction and erection have been observed in some animals during stimulation at all stages of the protocol. An important issue for patients with neurogenic bladder after spinal cord injury concerns vesico-ureteral reflux. Intravesical pressure rises between 50 and 60 cm H₂O for the 20 to 30 seconds necessary for complete bladder emptying while it stays

below 20 cm H₂O otherwise. The selective stimulation technique does not seem to cause ureteral reflux as confirmed by monthly voiding cystourethrogram.

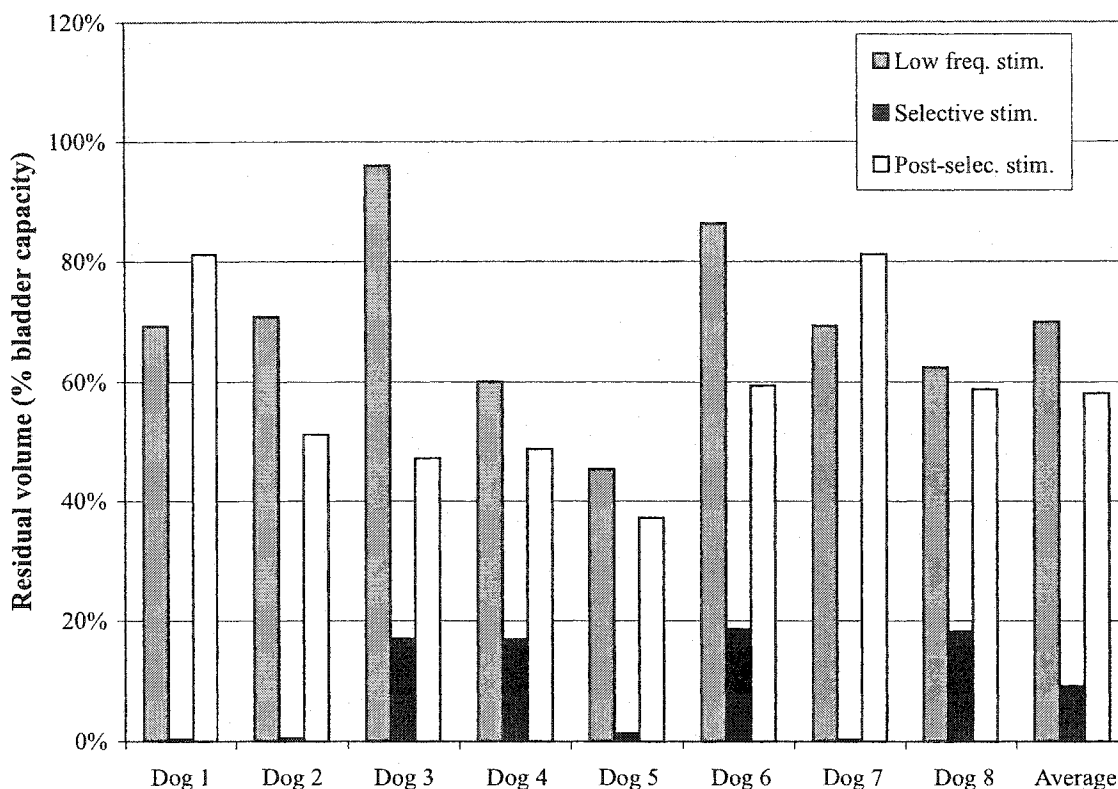


Figure 3.6: Residual volume for three stimulation steps: low-frequency only, selective and post-selective (for complementary information, see Table 3.2).

Erection has been known to reduce urinary flow during sacral root stimulation. Although selective stimulation by high-frequency inhibition does not prevent erection, it has usually been observed at the end of bladder evacuation, frequently lasted less than 1 minute and was not problematic for dogs. Also, in all cases, lower limb muscle contraction occurs as another side effect during sacral nerve excitation.

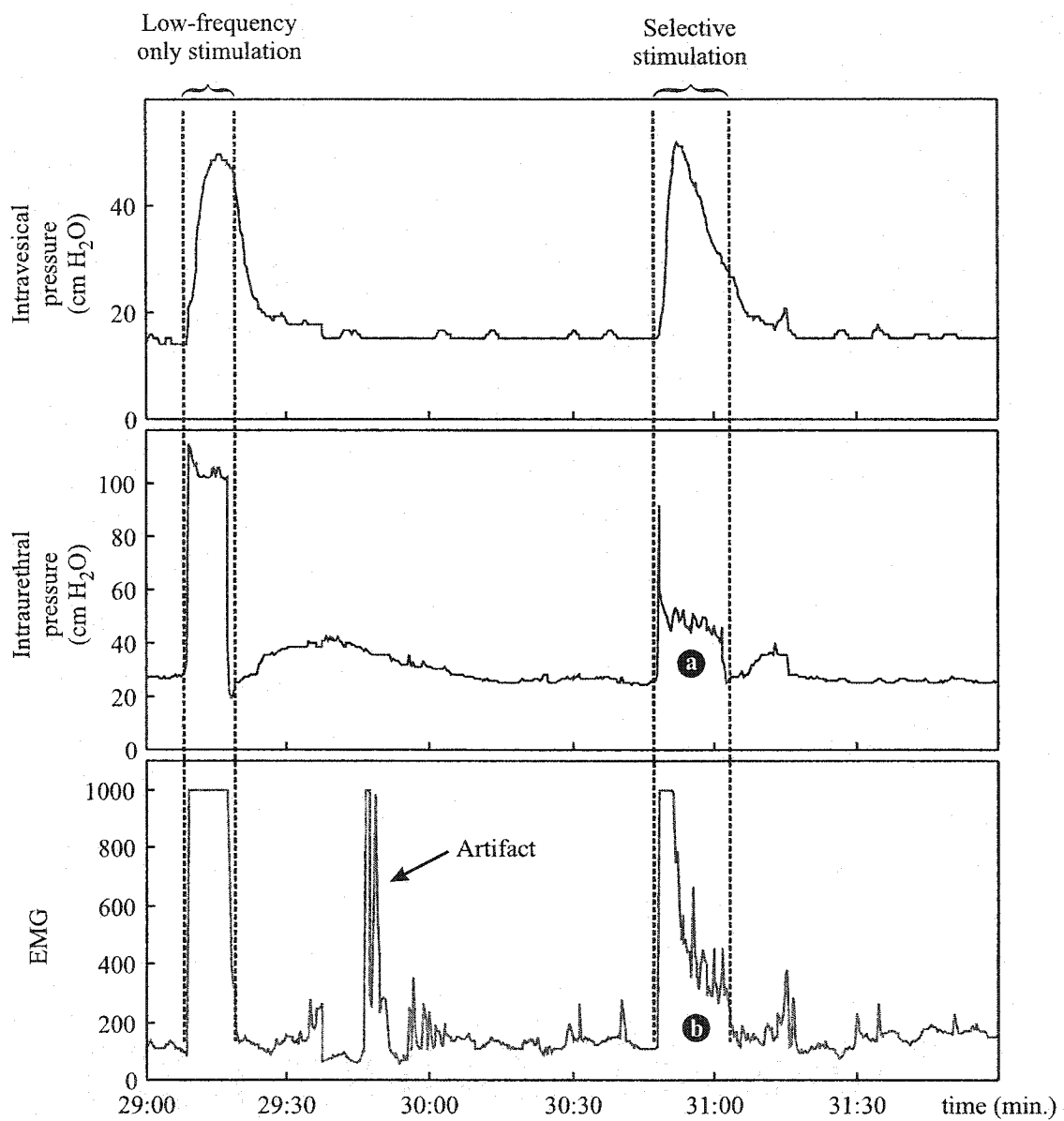


Figure 3.7: Intravesical and intraurethral pressures and EMG of the sphincter: a) lower intraurethral pressure allows micturition; b) reduced EMG coincides with lower intraurethral pressure.

3.2.6 Discussion

We presented in this article a new selective stimulation device together with its validation *in vivo* in dogs. Unilateral selective stimulation at the S2 level combining high-frequency stimuli dedicated to sphincter inhibition and low-frequency current pulses for detrusor muscle contraction leads to improved micturition. Selectivity is achieved by choosing the high-frequency pulses amplitude below the activation threshold of small nerve fibers leading to the bladder wall. This results in large nerve fibers stimulation and high-frequency fatigue of the sphincter leaving the detrusor muscle free for low-frequency stimulation. Some small fibers are activated by the high-frequency stimulus but the detrusor muscle high resistance to fatigue, combined to a lower sphincter fibers contraction, allows low-resistance voiding and therefore, lower pressure voiding with minimal residual volume of urine.

A previous attempt was reported by Brindley to fatigue the sphincter using sacral roots low-frequency stimulation. This study used 20 pulses per second stimuli to fatigue the sphincter for 3 minutes. The voltage was then increased to activate autonomic nerve fibers and produce maximal bladder contractions but the striated musculature contraction still impaired micturition. They concluded that low-frequency stimulation is inappropriate to fatigue the sphincter and leads to DSD.

The proposed implantable functional electrical stimulator and its external controller generate a wide range of reliable and precise stimuli. Together with the

implantation technique, the new system allowed effective voiding. An average residual urine volume of 9% of the bladder capacity was measured during chronic experiments on paralyzed dogs for which experiments lasted an average of eight months. Miniaturized full custom version of the implantable stimulator including a feedback loop to monitor the events surrounding the electrode-nerve contact is under development to improve the characterization of the proposed device.

The selective stimulation by high-frequency inhibition technique shows promising results in the dog which is an appropriate model for neurostimulation studies [24]. After spinal cord injury, a dog presents with vesicosphincteric dyssynergia which is quite similar to what is found in human. Therefore, the chronic experiments in dogs are important and future application in human will be possible soon.

The selective stimulation involves more energy than standard low-frequency only stimulation but the high-frequency train uses a low-current amplitude which results in a low-charge density due to short pulses duration. The charge quantity involved in the low-frequency train is evaluated to approximately $0.16 \mu C / phase$. If we consider the equivalent electrode surface (A) equal to $1 mm^2$, we obtain a charge density (δ) of $16 \mu C / cm^2$ for each phase.

$$\delta = \frac{I \cdot t}{A} = \frac{0.9mA \cdot 175\mu C}{1mm^2} \cong 16 \frac{\mu C}{cm^2 \cdot phase} \quad (1)$$

where I is the constant current amplitude and t is the pulse width duration.

Employed current amplitude and pulse width have been taken from Table 3.2.

Platinum electrodes are resistant to corrosion for charge densities up to 400 $\mu\text{C}/\text{cm}^2$ [13] while no significant neurological changes have been observed on the nerve due to chronic neurostimulation or reaction to the cuff electrode as demonstrated by nerve histology which will be the subject of a further publication.

Acknowledgments

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

3.3 Conclusion

Les chapitres suivants présentent deux nouveaux stimulateurs neuromusculaires. Le premier stimulateur, présenté au chapitre 4 se veut une version hautement intégrée du microstimulateur existant (MSN) utilisé pour les essais en phase chronique présentés dans ce chapitre. Il a été développé avant les nouvelles fonctions introduites par le NLB et le NSP. Ce nouvel implant fut spécifié en prévision de tests *in vivo* sur de petits animaux tel que le rat.

Le second stimulateur, présenté au chapitre 5, se veut une solution à un problème d'un autre ordre. Il vise l'utilisation d'un stimulateur externe et portable ayant des caractéristiques comparables aux systèmes implantables pour la sélection de fibres nerveuses adéquates à la stimulation sélective lors de la chirurgie. Le

système présentement utilisé possède, entre autres, un étage de sortie contrôlant une tension de stimulation tandis que les stimulateurs implantables contrôlent le courant de stimulation.

CHAPITRE 4

RÉALISATION ET TEST D'UN STIMULATEUR IMPLANTABLE INTÉGRÉ

4.1 Introduction

Le présent chapitre porte sur la réalisation d'un circuit intégré dédié à la récupération des fonctions vésicales. L'architecture du circuit est d'abord abordée, suivie de la description des modules numériques de réception des données, de contrôle et de génération des stimuli. Enfin, l'étage analogique de sortie est détaillé et la stratégie de vérification avant la production du circuit intégré, les tests effectués sur les circuits fabriqués et les résultats sont présentés.

Comme mentionné au chapitre 1, le stimulateur implantable est dédié à la récupération des fonctions vésicales par la stimulation sélective des nerfs sacrés contrôlant la vessie. Le signal de stimulation (figure 4.1), composé de deux trains d'impulsions bipolaires superposés, est défini par 8 paramètres, 4 pour chaque train d'impulsions, soit:

- l'amplitude des impulsions;
- la période d'un train d'impulsions;
- la durée d'une impulsion (positive et négative) et

- le délai entre l'impulsion positive et l'impulsion négative.

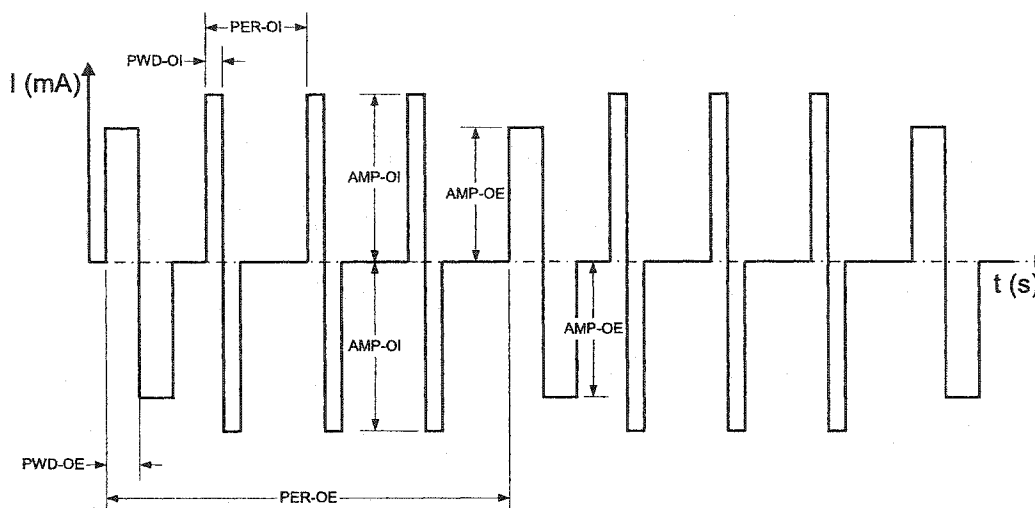


Figure 4.1: Forme du signal généré par le stimulateur implantable.

Le Tableau 4.1 présente la plage disponible pour chacun des paramètres de stimulation. La programmation des paramètres se fait à l'aide du contrôleur externe qui transmet l'ensemble de paramètres de façon sérielle, en même temps que l'énergie nécessaire au fonctionnement de l'implant, par un lien à couplage inductif.

Tableau 4.1: Plage disponible pour les paramètres de stimulation de l'implant.

Abréviation	Description	Plage	# bits
PER-OE	Période des impulsions de l'onde excitante	0 - 200 ms	16
AMP-OE	Amplitude des impulsions de l'onde excitante	0 - 3,0 mA	5
PWD-OE	Durée des impulsions de l'onde excitante	0 - 850 μ s	8
IPD-OE	Délai inter-impulsions de l'onde excitante	0 - 850 μ s	8
PER-OI	Période des impulsions de l'onde inhibitrice	0 - 200 ms	16
AMP-OI	Amplitude des impulsions de l'onde inhibitrice	0 - 3,0 mA	5
PWD-OI	Durée des impulsions de l'onde inhibitrice	0 - 850 μ s	8
IPD-OI	Délai inter-impulsions de l'onde inhibitrice	0 - 850 μ s	8

4.2 Design

4.2.1 Architecture

La figure 4.2 montre l'architecture générale du stimulateur implantable. Le signal radio-fréquence, modulé en amplitude et envoyé par le contrôleur externe, est capté par l'antenne et distribué à l'étage de démodulation et d'alimentation. Le démodulateur recouvre le signal numérique encodé en format Manchester tandis que l'étage de régulation récupère, à partir du lien inductif, l'énergie nécessaire au fonctionnement de l'implant et génère la tension d'alimentation de 3.3 volts. Le décodeur Manchester, réalisé principalement à partir de composants numériques, extrait les données et l'horloge du signal démodulé.

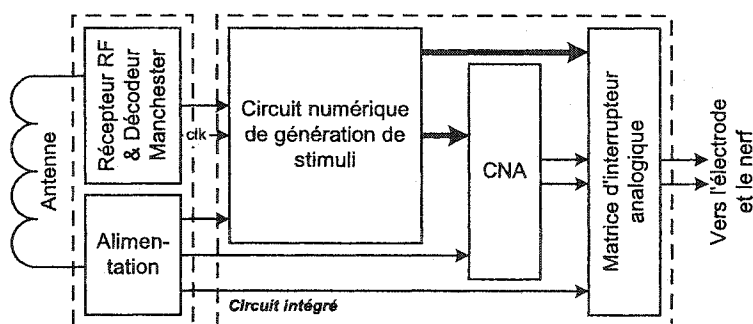


Figure 4.2: Schéma simplifié du stimulateur neuromusculaire implantable.

Les données sérielles sont ensuite mémorisées dans un registre à décalage pendant que le détecteur d'entête identifie le début de la séquence envoyée et initie le transfert des paramètres aux générateurs de stimuli. La sortie de chacun des générateurs est additionnée avant d'être convertie en un signal analogique bipolaire.

laire par le convertisseur numérique à analogique (CNA).

4.2.2 Alimentation de l'implant et réception RF

Les travaux de Robin ^[33] proposent un régulateur de tension linéaire pour stabiliser la tension et un détecteur d'enveloppe accompagné d'un filtre pour récupérer le signal RF. Ces circuits, conçus pour un procédé BiCMOS de $0,8\ \mu\text{m}$, utilisent à la fois des transistors MOS et bipolaires. Comme la réalisation actuelle du stimulateur implantable vise un procédé CMOS de $0,35\ \mu\text{m}$, il est nécessaire de revoir les designs proposés pour une réalisation complètement intégrée du stimulateur implantable. Cette revue fera l'objet de travaux ultérieurs.

4.2.3 Module numérique de réception et de contrôle

La partie numérique du stimulateur implantable est principalement constituée d'un registre à décalage avec détection d'entête, de deux instances du module de génération de stimuli et d'un module d'addition (figure 4.3). Quelques modules de test sont également présents mais ne sont pas identifiés sur le schéma simplifié par souci de clarté.

La séquence des paramètres est reçue par le registre à décalage qui verrouille les données lorsqu'une entête identifiant une opération de stimulation ou de test est détectée. Dans le premier cas (stimulation), les modules de génération des stimuli utilisent les paramètres définissant la période, la largeur des impulsions, le

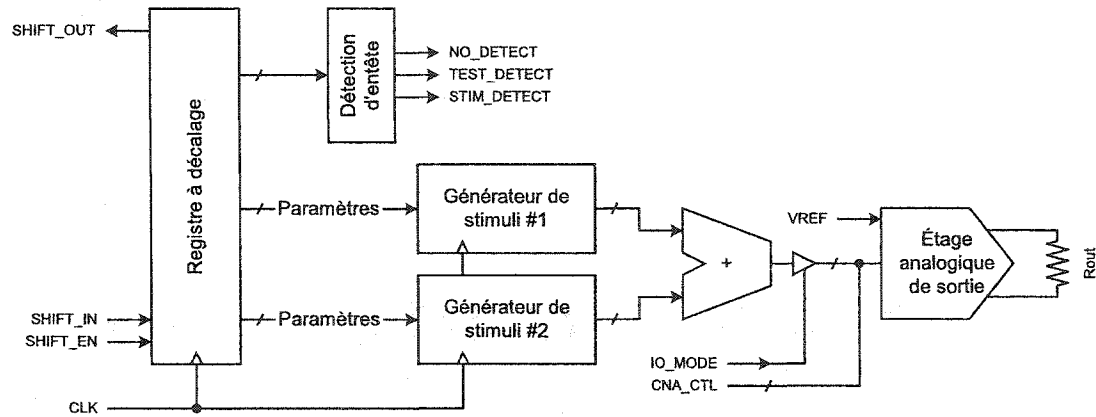


Figure 4.3: Schéma simplifié du circuit intégré dédié.

délai inter-impulsion et l'amplitude des ondes excitante et inhibitrice pour créer le signal de stimulation voulu. Les sorties des modules de génération des stimuli sont ensuite additionnées et le résultat est envoyé au convertisseur numérique à analogique (CNA). Dans le second cas (test), une rampe bipolaire est générée au niveau de l'additionneur et est envoyée au CNA. Ce mode permet de tester les 32 niveaux de courant et la bidirectionnalité de l'étage analogique de sortie.

4.2.4 Étage analogique de sortie

L'étage analogique de sortie est constitué d'un convertisseur numérique à analogique de 5 bits et d'un étage de sortie bipolaire permettant de choisir le sens du courant de stimulation, soit d'injecter ou d'extraire des charges des tissus nerveux. Les circuits du CNA et de sélection de la polarité du courant sont présentés à la figure 4.4.

La partie analogique présentée dans ce mémoire et réalisée par Jean-Charles

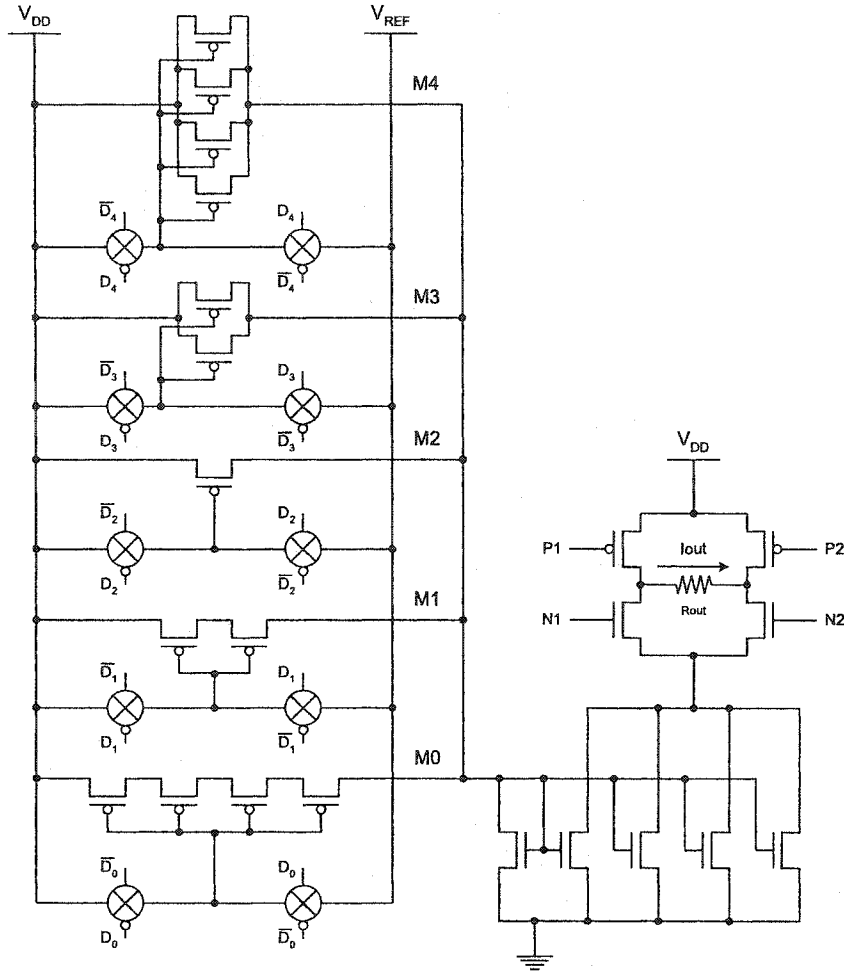


Figure 4.4: Schéma électrique de l'étage analogique de sortie (CNA et étage de sortie bipolaire).

Voghell, a été conçue spécifiquement pour le stimulateur implantable dédié à la récupération des fonctions urinaires. Le CNA de type exponentiel, utilisé initialement par St-Amand [42], [43] et ensuite par Bourret [6], occupe une faible surface et offre une bonne linéarité. Les sources de courant sont pondérées de façon binaire (M0-M4) et commandées indépendamment par les signaux de contrôle de l'amplitude ($D_4 - D_0$). Les sources sont réalisées à partir de transistors de même

dimension mis en série ou en parallèle de façon à obtenir des courants proportionnels au poids de chacun des bits (puissances de 2). La géométrie unique des transistors permet d'obtenir un meilleur contrôle sur les rapports de courants en évitant les erreurs de modélisation des transistors de taille différente, tandis que la mise en série pour la division du courant de référence, et en parallèle pour la multiplication, permet de réduire la surface du CNA (par opposition à des transistors mis en série ou en parallèle uniquement).

Chacun des signaux de contrôle attaque d'abord un multiplexeur analogique (identifié, sur le schéma, par les portes de transmission) permettant de choisir entre la tension d'alimentation (V_{DD}) ou la tension de référence (V_{REF}) pour alimenter la grille des transistors utilisés comme sources de courant pondérées. Cette architecture offre la possibilité de choisir une tension de référence supérieure à 0 volt et permet de s'assurer que les transistors composant les sources de courant pondérées opèrent en saturation, améliorant ainsi la linéarité du convertisseur. La somme des courants des sources pondérées est ensuite multipliée par un miroir de courant qui permet de conserver une charge constante à la sortie du CNA. L'utilisation d'un miroir de courant unique assure également que le courant reste le même peu importe la polarité choisie, minimisant ainsi la composante continue du signal bipolaire. Finalement, l'architecture en «H» autour de la charge (R_{out}) permet de choisir le sens du courant de stimulation en commandant, par pair, les signaux de contrôle P1, P2, N1 et N2.

Une attention particulière à également été portée sur la disposition des transistors du CNA (figure 4.5), disposés autour d'un centre de masse commun pour réduire les effets des variations locales du procédé de fabrication ($0,35\ \mu\text{m}$).

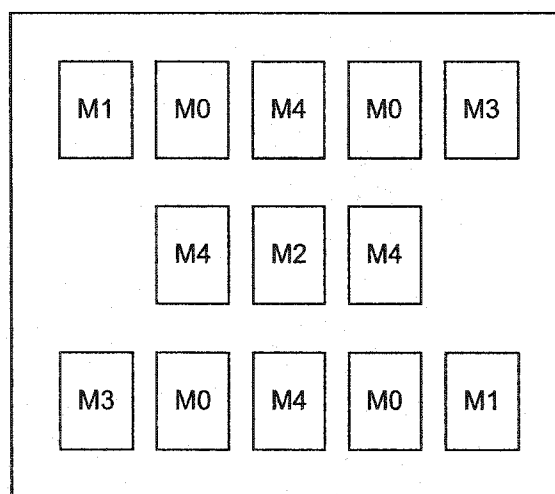


Figure 4.5: Géométrie de type centre de masse commun utilisée pour le CNA.

La figure 4.6 montre le dessin des masques de l'étage de sortie analogique et identifie la disposition des différentes parties de la source de courant bipolaire. On peut remarquer, entre autre, que l'espace occupé pourrait être réduit significativement. Cependant, comme le stimulateur implantable contient une logique réduite et que la surface du dé de silicium est tributaire du nombre de plots, un effort de réduction de la surface occupée par le CNA n'a pas été nécessaire.

4.3 Vérification, synthèse et dessin des masques

La vérification fonctionnelle de la partie numérique a été réalisée en utilisant un banc d'essai (testbench) contrôlant les entrées du stimulateur implantable. Une

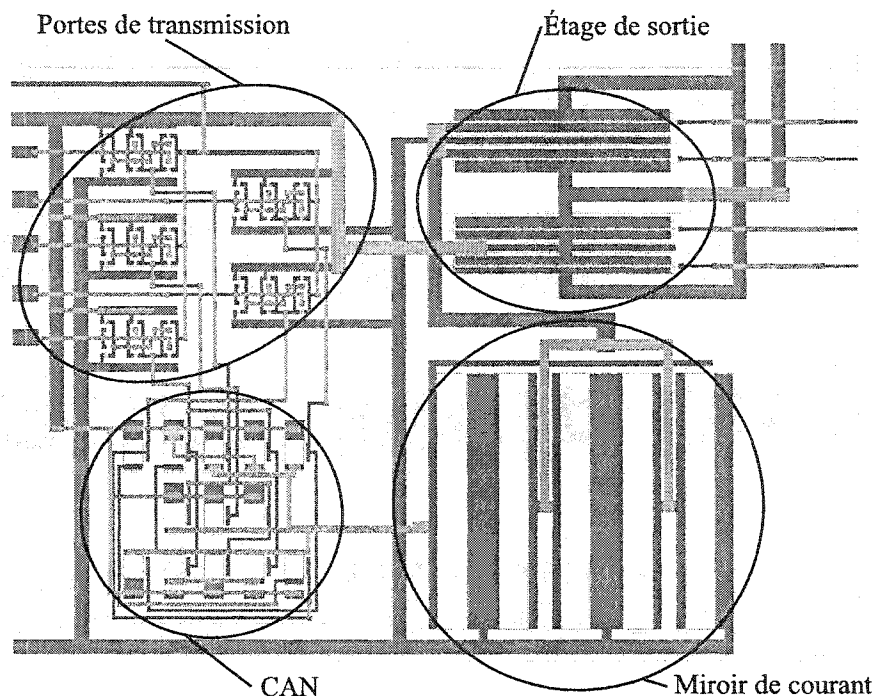


Figure 4.6: Dessin des masques de la source de courant.

séquence de programmation sérielle est d'abord envoyée et la vérification du fonctionnement se fait visuellement à partir de l'amplitude (numérique) du signal de sortie. La compilation du VHDL pour la vérification fonctionnelle et la simulation est faite à l'aide de l'outil ModelSim. Cette méthode de vérification est longue et fastidieuse car le code VHDL du banc d'essai (testbench) doit être modifié pour chacun des tests et les risques d'erreurs demeurent élevés après ce type de vérification. Une méthodologie plus robuste est discutée dans la conclusion du chapitre.

Du côté de la vérification de la partie analogique (source de courant), les simulations ont été réalisées par Jean-Charles Voghell à partir du schéma et également à partir de la liste d'interconnexions extraite du dessin des masques pour valider

le fonctionnement adéquat. Les simulations réalisées sur la liste d'interconnexions extraite (extracted netlist) permettent de s'assurer que la taille des transistors est conforme au schéma et que les éléments parasites, principalement les capacités, ne nuisent pas outre mesure au bon fonctionnement de la source de courant.

La réalisation du dessin des masques de la partie numérique suit la méthodologie introduite dans le guide d'initiation à la réalisation de design numérique de la société canadienne de micro-électronique (SCM ^[15]). La synthèse de la partie numérique a été réalisée à partir de l'interface graphique de l'outil Synopsys en utilisant la librairie de cellules «wcells» pour la technologie 0,35 μm de TSMC (Taiwan Semiconductor Manufacturing Company). L'analyse statique de délais (ASD) a également été réalisée à l'aide de l'interface graphique des outils associés à Synopsys.

Après la synthèse, un arbre de distribution du signal d'horloge est normalement inséré pour minimiser les délais entre chacun des éléments mémoires synchrones (bascules D) pour éviter les problèmes de temps de maintien. Cette opération a malheureusement été omise pour le présent design, en raison de la méthodologie présentée dans le guide d'initiation au design numérique de la SCM et du court délai pour réaliser le projet.

Dans le design de circuit dédié à la production, une chaîne de balayage permettant de tester de façon extensive la partie numérique est normalement insérée. Cette opération, réalisée au cours de la synthèse, remplace les bascules D conven-

tionnelles par des bascules D de type «scan» permettant de créer un ou plusieurs registres à décalage comprenant l'ensemble des registres du design. Lors d'un test, il est possible de charger les registres avec une valeur prédéterminée ou de récupérer le résultat après un cycle d'horloge en fonctionnement normal. La valeur initiale et le résultat constitue un vecteur de test. La chaîne de balayage peut également être utilisée pour le diagnostic du prototype. Toutefois, l'insertion d'une chaîne de balayage augmente la surface d'un circuit et apporte peu lors du design d'un prototype de faible complexité, pour lequel la testabilité ne représente pas un élément clé nécessaire à la réussite d'un projet et où des tests fonctionnels permettent de localiser rapidement un problème. Vu la taille réduite du circuit numérique et puisque le présent projet n'est pas destiné à un volume de production important, la chaîne de balayage n'a pas été insérée.

L'ajout de la partie analogique a nécessité quelques opérations non documentées dans le guide d'initiation de la SCM. Le design de la partie analogique n'étant complété que tardivement, il fut nécessaire de générer une vue abstraite de la source de courant pour l'intégrer au dessin des masques. Cette vue simplifiée conserve les caractéristiques nécessaires au placement et au routage du design soit, les dimensions, la position des ports d'entrée et de sortie ainsi que les couches de métal utilisées pour les interconnexions et les zones d'exclusions de métal.

Le dessin des masques résultant comprend 24 plots et occupe une surface de $1750\text{ }\mu\text{m}$ par $2090\text{ }\mu\text{m}$, soit $3,65\text{ mm}^2$. Comme mentionné précédemment, la surface

du dé de silicium est délimitée par le nombre de plot (I/O bound) car la surface à l'intérieur des plots n'est pas complètement occupée par les parties numérique et analogique. Dans le but de respecter la densité minimale pour chacune des couches de métal, des rectangles métalliques ont été insérés dans les espaces vides après le routage.

4.4 Test du circuit intégré

Parmi les 20 exemplaires fabriqués, 10 échantillons ont été encapsulés dans des boîtiers DIP de 40 broches et 10 échantillons demeurent non-encapsulés pour une utilisation éventuelle dans un prototype miniaturisé du stimulateur neuromusculaire implantable.

Les tests préliminaires ont permis d'identifier quelques problèmes de conception. Dans les conditions nominales, soit avec une alimentation de 3,3 volts, le registre à décalage utilisé pour mémoriser le mode de fonctionnement (entête) et les paramètres de stimulation montre un problème de temps de maintien (hold time). Le registre à décalage a une longueur de 77 bits et après l'initialisation à zéro de tous les bits, moins de 77 coups d'horloge sont nécessaires pour voir apparaître un '1' à la sortie du registre. La réduction de la tension d'alimentation jusqu'à environ 2,5 volts (la tension varie légèrement d'un échantillon à l'autre) permet d'obtenir un fonctionnement adéquat de la partie numérique. Une analyse plus approfondie a permis de relier ce problème à une mauvaise distribution

de l'horloge, qui est mise en évidence en premier lieu dans le registre à décalage à cause de l'absence de logique combinatoire entre la sortie d'une bascule D et l'entrée de la bascule D adjacente. En utilisant une alimentation de 2,5 volts, les tests fonctionnels exercés sur la partie numérique se sont tous conclus avec succès.

Le mode de test de la partie numérique permet de générer une rampe bipolaire à l'entrée du CNA. Les premières lectures à la sortie de la source de courant en utilisant ce mode ont rapidement montré un problème d'interface entre les parties numérique et analogique. Une erreur de conception a eu pour effet d'inverser le branchement des signaux de contrôle de l'étage de sortie bipolaire permettant de choisir le sens du courant de stimulation. Ainsi, la partie numérique active les transistors contigus permettant au courant de rejoindre la masse sans traverser la charge (R_{out}). Un branchement adéquat aurait jumelé les transistors alternés forçant le courant à travers la charge. Les tests sur la partie analogique ont été réalisés en isolant la source de courant de la partie numérique, un mode de fonctionnement prévu lors de la conception.

La figure 4.7 montre le courant de sortie mesuré en fonction de l'entrée binaire pour différentes tensions de référence dans une charge de 1,0 k Ω . Vu le type d'analyse, les données présentées proviennent d'un seul échantillon, choisi dans la moitié inférieure en terme de linéarité. Lorsque la tension de référence est nulle, la pente de la rampe est élevée et on peut clairement voir la saturation autour de 2 mA. L'utilisation d'une tension de référence non-nulle améliore la linéarité de la

source de courant mais réduit la pente de la rampe et limite le courant maximal.

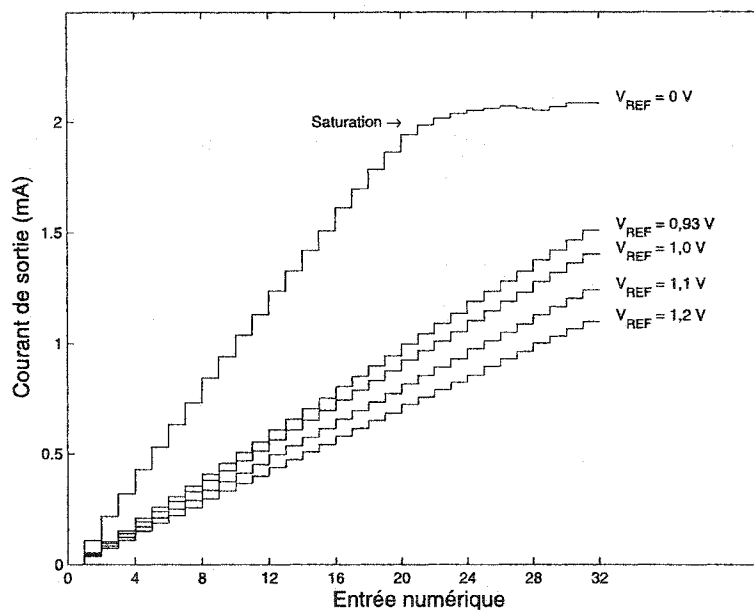


Figure 4.7: Courants de sortie mesurés en fonction de différentes tensions de référence.

La figure 4.8 présente les non-linéarités différentielle et intégrale pour une tension de référence de 0,93 volt fournissant à un courant maximal de sortie de 1,5 mA. Cette tension de référence donne une non-linéarité différentielle maximale de 0,17 LSB et une non-linéarité intégrale maximale de 0,52 LSB. L'utilisation d'une tension de référence de 1,2 volt réduit le courant de sortie maximal à 1,1 mA mais, comme prévu, améliore significativement la linéarité de la source de courant tel que montré par la figure 4.9. Dans ces conditions, la non-linéarité différentielle maximale est de 0,16 LSB tandis que la non-linéarité intégrale maximale tombe à 0,41 LSB.

Les performances de la source de courant présentée dans ce chapitre sont mod-

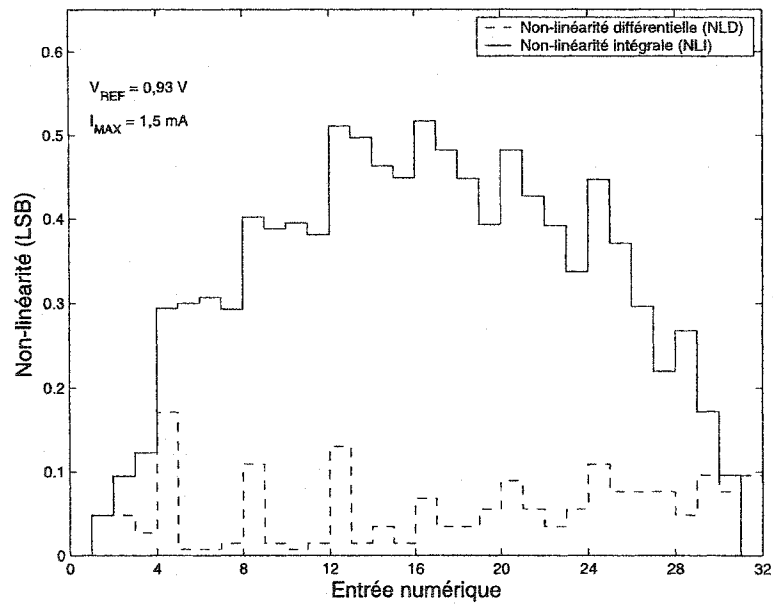


Figure 4.8: Non-linéarités différentielle et intégrale du convertisseur pour une tension de référence de 0,93 volt.

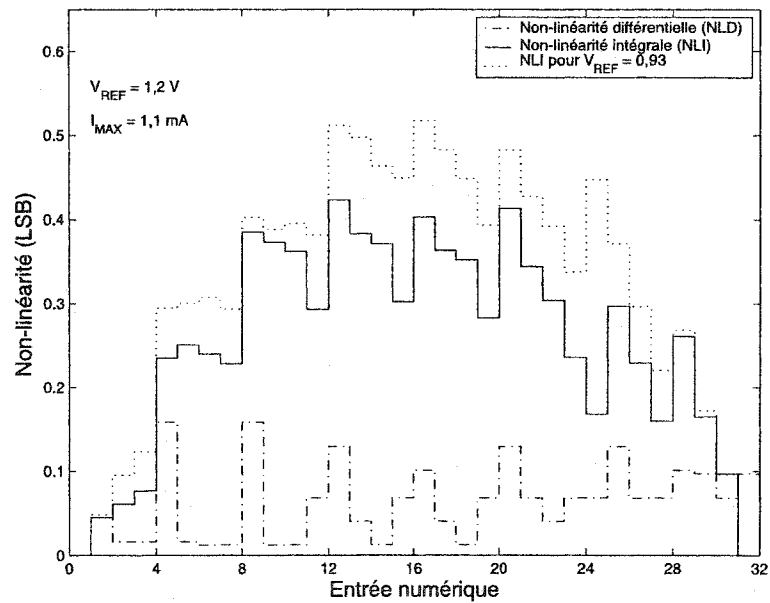


Figure 4.9: Non-linéarités différentielle et intégrale du convertisseur pour une tension de référence de 1,2 volt.

estes lorsque comparées avec les performances des sources de St-Amand et de Bourret. Pour quantifier l'ensemble des résultats présenté, nous utilisons le facteur de qualité Q , à minimiser, défini par St-Amand selon l'équation:

$$Q = A \cdot L \quad (4.1)$$

où A est la surface occupée par la source de courant (en μm^2) et L le facteur de linéarité du courant de sortie défini par:

$$L = NLI + NLD \quad (4.2)$$

où NLI et NLD sont respectivement les non-linéarités intégrale et différentielle maximales du courant de sortie, exprimées en LSB.

Tableau 4.2: Comparaison des caractéristiques des sources développées par St-Amand (1995), Bourret (1999) et de la nouvelle version proposée.

Circuit	A	L	Q
St-Amand (CMOS 3,0 μm)	1698	1,10	1867,8
St-Amand (CMOS 1,2 μm)	678	0,54	366,1
Bourret (BiCMOS 0,8 μm)	1765	0,493	870,1
Voghell, $V_{REF} = 0,93$ volt (CMOS 0,35 μm)	16942	0,571	9673,9

Bien que la source de courant soit réalisée à l'aide du procédé CMOS de 0,35 μm , sa taille est significativement supérieure aux sources proposées par St-Amand et Bourret. Ce paramètre influence grandement le facteur de qualité défini par St-Amand et la classe au dernier rang des designs proposés. La source proposée

par Voghell est toutefois la seule à utiliser une alimentation de 3,3 volts tout en tentant de conserver un courant de sortie maximal relativement élevé dans la charge nominale de 1 k Ω . Il est également nécessaire de mentionner que le design atteint des performances comparables du point de vue de la linéarité aux autres designs utilisant une technologie CMOS.

4.5 Conclusion

Il est maintenant nécessaire de revenir sur certains points mentionnés au cours du chapitre. Le banc d'essais utilisé pour la vérification de la partie numérique nécessite une modification de la description VHDL pour chacun des tests. Un banc d'essais plus versatile permettrait de choisir les paramètres de test à l'aide d'un fichier d'entrée et générerait un fichier de sortie montrant le fonctionnement du stimulateur. Ce fichier de sortie pourrait ensuite être comparé avec un fichier de référence défini pour le test en cours. Il serait ainsi possible d'effectuer tous les tests nécessaires à la vérification systématique complète de la partie numérique sans modifier le banc d'essais.

Une méthodologie d'analyse statique des délais plus robuste suivant la synthèse de la description VHDL et le placement aurait également permis de constater les problèmes de temps de maintien (hold time) dans le registre à décalage destiné à la mémorisation des paramètres de stimulation. Cette méthodologie doit tenir en compte de la longueur des fils et de la capacité parasite associée.

Bien que les performances de la source de courant soient limitées lorsque comparées aux designs précédents, elle répond aux exigences de l'application pour les spécifications nominales. Toutefois, le faible courant maximal de sortie et la saturation rapide pour une charge supérieure à $1\text{ k}\Omega$ limitent l'utilisation du nouveau convertisseur présenté dans ce mémoire. Le courant maximal de $1,1\text{ mA}$, pour une linéarité optimale, laisse une marge de manoeuvre très restreinte lors de l'optimisation des paramètres de stimulation pendant la qualification d'un prototype ou la validation de la méthode de stimulation.

CHAPITRE 5

RÉALISATION D'UN STIMULATEUR EXTERNE PORTABLE

5.1 Introduction

Le chapitre 4 était consacré au microstimulateur miniaturisé dédié à la récupération des fonctions urinaires. Toutefois, les procédures menant à la localisation de la racine sacrée produisant une contraction maximale de la vessie et les expérimentations de stimulation sélective utilisent des outils commerciaux ayant peu de points de référence avec l'implant décrit. Le stimulateur électrique commercial SD9 de Grass Medical Instruments est présentement utilisé lors des chirurgies. Il offre un contrôle de la tension tandis que le microstimulateur implantable délivre du courant. Le stimulateur SD9 est également limité du point de vue de la fréquence et du nombre de trains d'impulsions simultanés. Pour uniformiser les outils et mieux gérer les données recueillies lors des chirurgies, le prototype d'un nouveau stimulateur externe (figure 5.1) a été réalisé et est décrit dans le présent chapitre.

5.2 Architecture générale

Comme mentionné précédemment, le stimulateur externe doit posséder des caractéristiques semblables au système implantable soit, un contrôle numérique des paramètres, un étage de sortie en mode courant et la possibilité de superposer

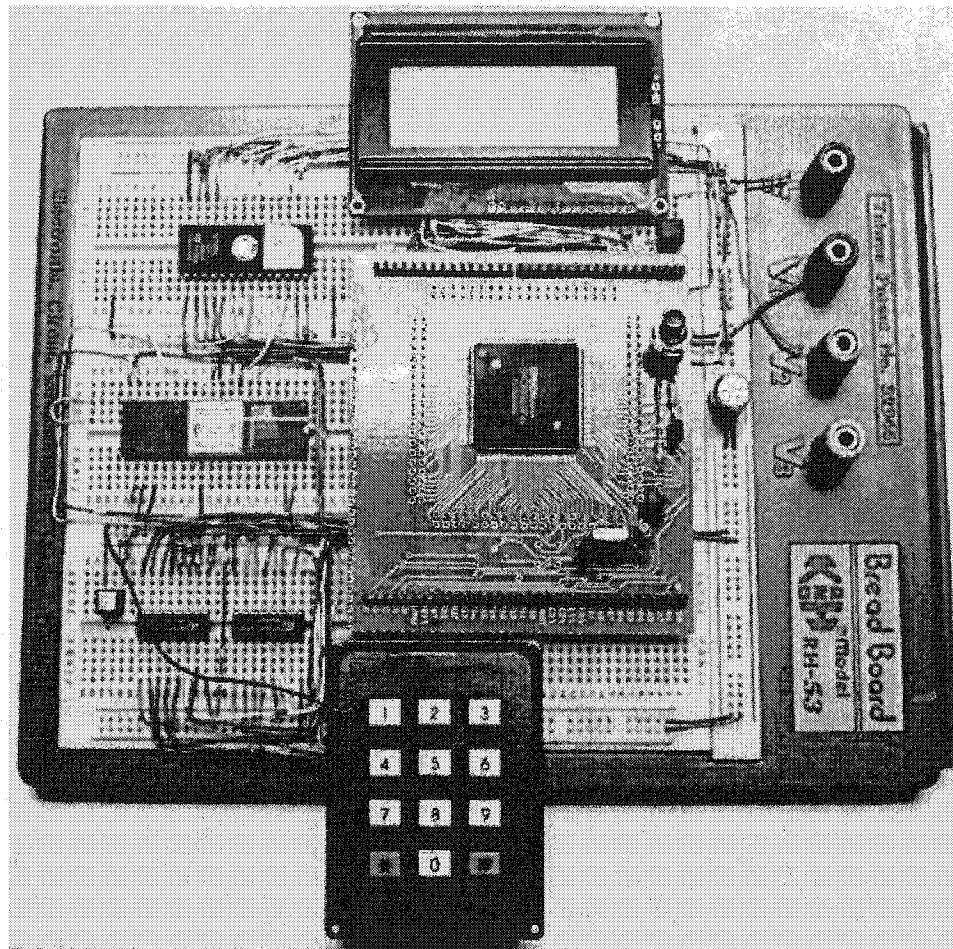


Figure 5.1: Photo du prototype du stimulateur externe portable.

deux trains d'impulsions. D'autres objectifs sont également visés par le design de ce nouvel outil, soit:

- la création d'une interface utilisateur flexible et facile d'utilisation pour modifier les paramètres de stimulation;
- la portabilité;
- le faible encombrement et

- la mise à jour ou l'ajout de fonctions au design existant par programmation uniquement.

L'architecture du stimulateur externe (figure 5.2) est basée sur le microcontrôleur EPSON E0C88316 et utilise le circuit intégré (CI) décrit au chapitre 4. Le microcontrôleur gère l'interface usager, la validation des paramètres de stimulation et le contrôle du circuit dédié tandis que le CI génère les formes d'onde et les courants de stimulation désirés. Le schéma électrique est fourni à l'annexe II.1. Le système comprend également une mémoire morte de 64 kbits pour le programme à exécuter, une mémoire morte effaçable de 1 kbits pour conserver quatre ensembles de paramètres après la coupure de l'alimentation, un clavier et son encodeur, un écran LCD de 4 lignes par 16 caractères avec contrôleur intégré et un voyant lumineux (LED) pour indiquer qu'une stimulation est en cours.

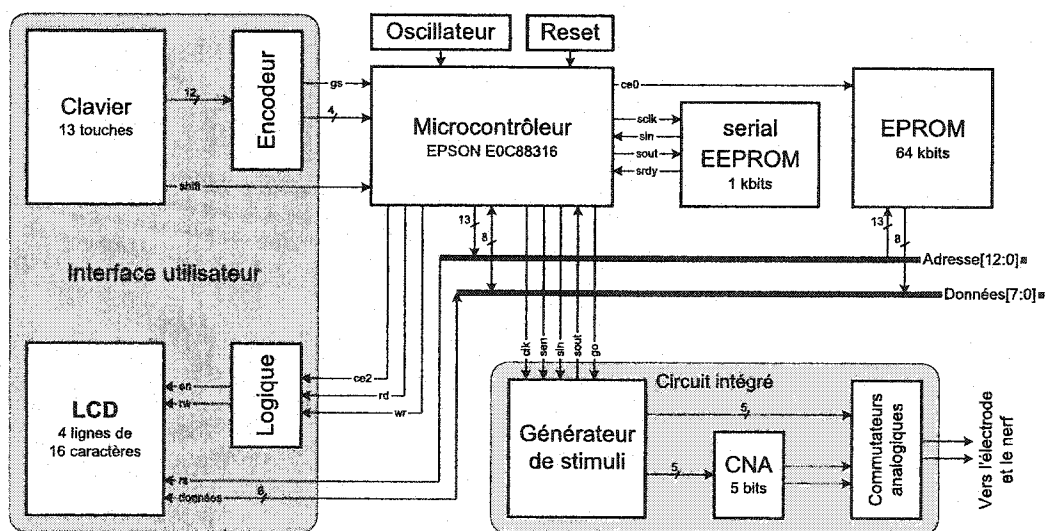


Figure 5.2: Architecture générale du stimulateur externe.

5.3 Interface utilisateur

Comme mentionné précédemment, l'interface utilisateur comporte un clavier de 13 touches (chiffres de 0 à 9, étoile [*], carré [#] et la touche pour accéder aux caractères étendus [shift]) et un écran LCD gérés par le microcontrôleur. Le programme principal, dont l'algorithme est donné à la figure 5.3, initialise le système et attend que l'utilisateur entre une commande à l'aide du clavier, interprète cette commande et met l'affichage à jour avant de retourner en mode d'attente.

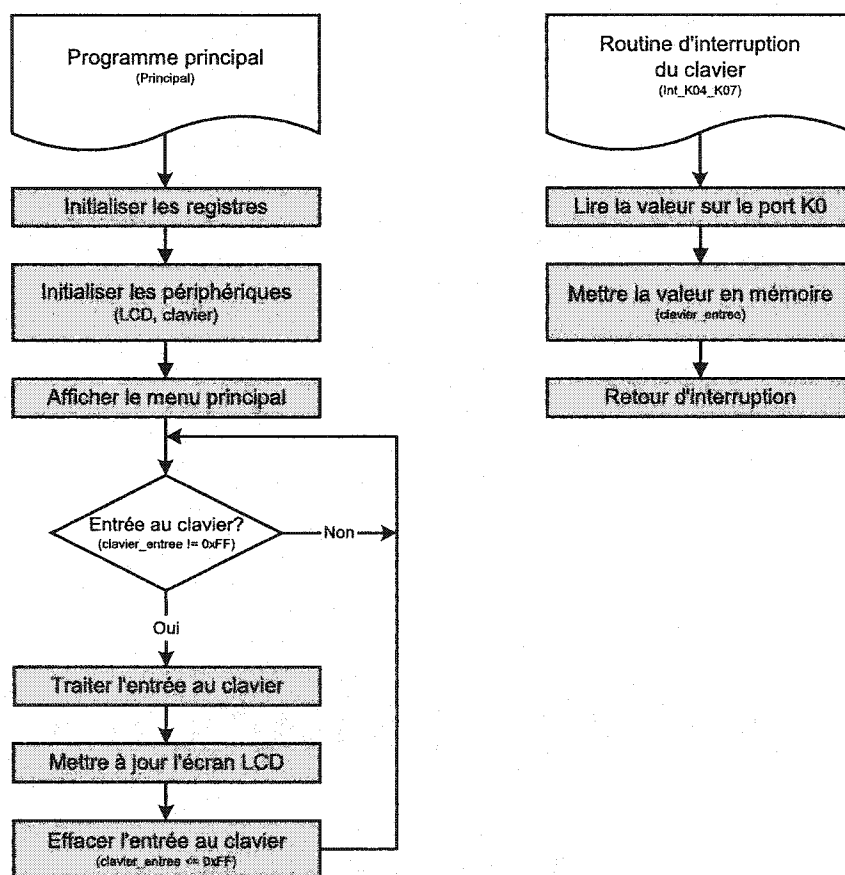


Figure 5.3: Algorithmes du programme principal du stimulateur externe et de la routine d'exception d'entrée au clavier.

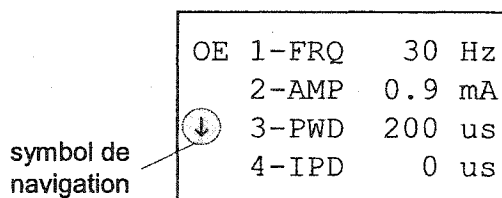
Après la période d'initialisation, le stimulateur externe offre trois options à l'utilisateur :

1. Vérification et modification des paramètres de stimulation;
2. Gestion des ensembles de paramètres en mémoire ou
3. Stimulation neuromusculaire.

Chacun de ces choix mène à un sous-menu explicite.

5.3.1 Modification des paramètres

L'option «Paramètres» permet de visualiser la valeur des 8 paramètres de stimulation (par groupe de 4) et de la modifier à l'aide du clavier numérique. Les paramètres sont regroupés dans les fenêtres OE (onde excitante) et OI (onde inhibitrice) et le passage d'une fenêtre à l'autre se fait à l'aide des flèches vers le haut [shift+2] et vers le bas [shift+8], respectivement, tel qu'indiqué par les symboles de navigation à la figure 5.4. La touche étoile permet, en tout temps, de revenir au menu précédent.



OE 1-FRQ	30 Hz
2-AMP	0.9 mA
3-PWD	200 us
4-IPD	0 us

Figure 5.4: Lorsque l'espace d'affichage est insuffisant, la navigation se fait à l'aide des flèches.

Une fois modifiée, la nouvelle valeur est vérifiée en fonction de l'intervalle de validité du paramètre courant uniquement. Le paramètre nouvellement modifié n'est confronté aux autres paramètres de stimulation qu'en passant au mode stimulation à partir du menu principal. Ainsi, l'ordre dans lequel les valeurs de paramètre sont modifiés n'a pas d'importance.

5.3.2 Gestion des espaces mémoires

L'option «Mémoire» permet de récupérer un ensemble de paramètres sauvegardés précédemment ou de sauvegarder l'ensemble actuel dans l'un des 4 espaces mémoires. Lors du chargement d'un ensemble de paramètres, l'utilisateur peut visualiser la valeur des paramètres à l'aide des touches de navigation. Lorsque le choix s'arrête sur un espace mémoire particulier (A, B, C ou D), l'utilisateur charge l'ensemble de paramètres à l'aide de la touche [#]. Lors de la sauvegarde, l'utilisateur n'a pas à naviguer parmi les ensembles de paramètres déjà en mémoire, il ne fait qu'entrer l'espace mémoire dans lequel il désire sauvegarder l'ensemble de paramètres actifs.

5.3.3 Stimulation neuromusculaire

La dernière option fait passer le système en mode «Stimulation». Comme mentionné précédemment, c'est à ce moment que l'onde de stimulation est construite et que la validité de l'ensemble de paramètres est vérifiée. L'utilisateur peut entrer

la durée de stimulation désirée ou utiliser la valeur par défaut (30 secondes). Une fois que la stimulation est activée, un indicateur lumineux (LED) s'allume et un compte à rebours indique le temps de stimulation restant. L'utilisateur peut interrompre à tout moment la stimulation en cours en appuyant sur une touche du clavier, même si la durée de stimulation ne s'est pas complètement écoulée.

5.3.4 Sous-routine de rafraîchissement de l'affichage

Les caractères constituant les fenêtres pour l'ensemble de l'affichage du stimulateur externe sont entrés dans un tableau (annexe II.2) où les paramètres modifiables sont identifiés par le caractère '\$'. Un second tableau de même dimension contient, aux positions marquées par le caractère '\$', les 8 bits les moins significatifs (LSB) de l'adresse à laquelle se trouve la valeur du paramètre modifiable dans la mémoire vive (RAM). Comme ces variables sont en mémoire vive (RAM), les 8 bits les plus significatifs (MSB) sont fixes et connus. Cette structure permet de simplifier considérablement la sous-routine de rafraîchissement de l'affichage pour l'écran LCD (figure 5.5). Cette sous-routine utilise deux index (IX et IY) pointant sur le premier caractère de la fenêtre à afficher dans les deux tableaux mentionnés précédemment et un marqueur de position (L) marquant la position du caractère dans la fenêtre.

La sous-routine de rafraîchissement de l'affichage de l'écran LCD débute par positionner le curseur sur la première ligne de l'écran et par la lecture du caractère à l'adresse (IX+L). S'il s'agit d'une variable (identifiée par '\$'), elle accède la valeur

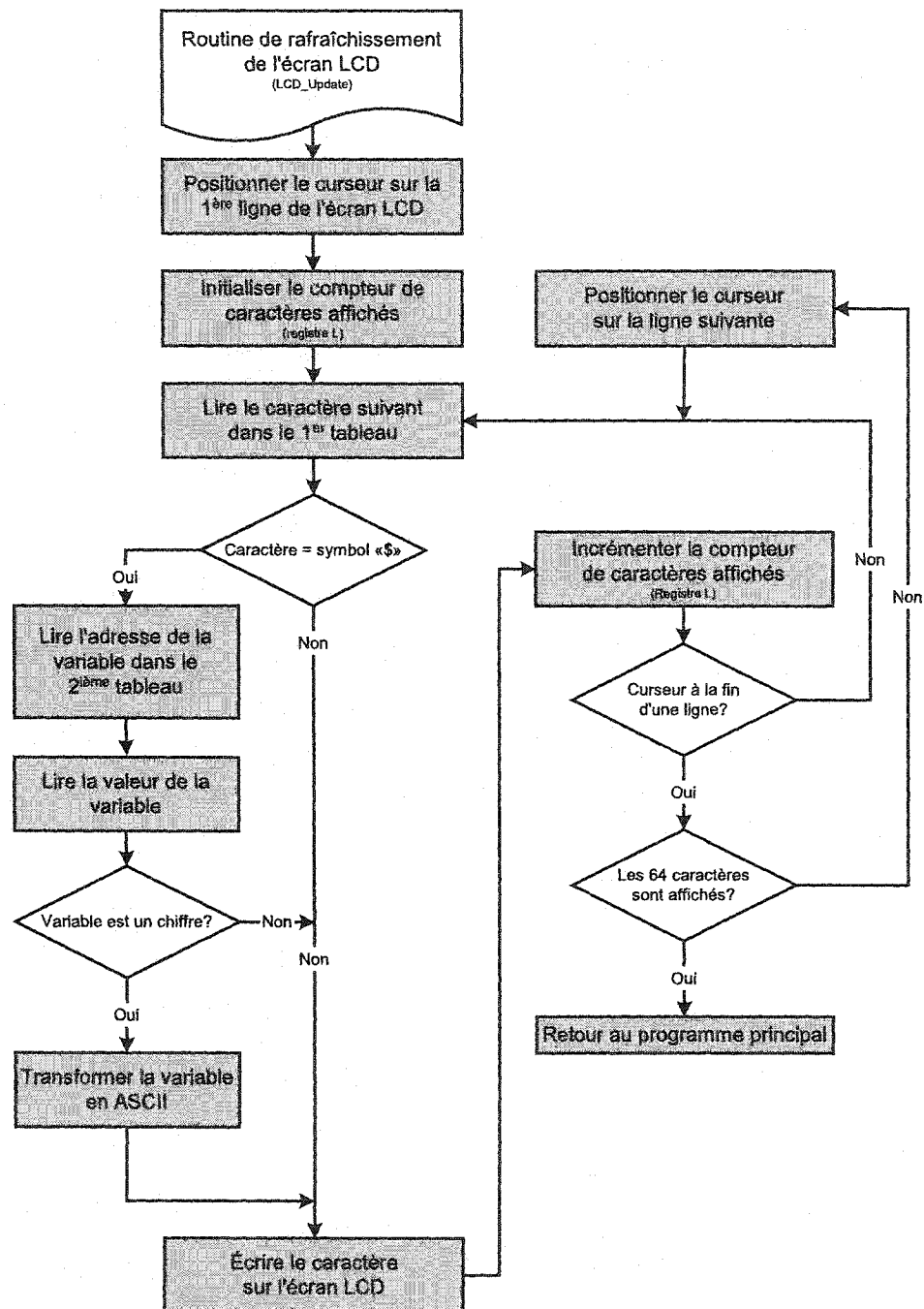


Figure 5.5: Algorithme de la sous-routine de rafraîchissement de l'affichage de l'écran LCD.

par l'intermédiaire du second tableau (IY+L) et transforme la valeur binaire en caractère ASCII. Le caractère, ou la variable, est ensuite affiché et le marqueur de position est incrémenté. Si le curseur est en fin de ligne (i.e. le marqueur est égale à 16, 32 ou 48), il est déplacé au début de la ligne suivante. La sous-routine est répétée à partir de la lecture du caractère jusqu'à ce que les 64 caractères composant chacune des fenêtres soient affichés.

Les algorithmes du programme principal et de la sous-routine de rafraîchissement de l'écran LCD sont les seuls détaillés dans le corps du mémoire, les autres algorithmes étant de complexité réduite. Le code assembleur du stimulateur externe peut être trouvé à l'annexe II.2.

5.4 Génération des stimuli et étage analogique de sortie

Une fois que l'ensemble de paramètres est choisi et que l'utilisateur entre dans le mode de stimulation neuromusculaire, les données sont traitées afin d'être utilisées par le module de génération des stimuli, décrit au chapitre précédant. L'utilisation de ce circuit a permis de réduire la complexité du design du stimulateur externe et de remplir un des objectifs du prototype, soit d'uniformiser les types d'étage de sortie des stimulateurs neuromusculaires utilisés lors des différentes phases de l'expérimentation.

5.5 Vérification et tests

La vérification du stimulateur externe reste limitée puisqu'aucun simulateur modélisant le comportement du microcontrôleur en fonction du code source n'est disponible. La vérification du module de génération des stimuli et de l'étage analogique de sortie est élaborée au chapitre précédent.

Un ensemble de tests de fonctionnement permettant de vérifier les limites d'opération a été réalisé avec succès sur le stimulateur externe. Le passage d'un menu d'affichage à un autre, la sauvegarde en mémoire ainsi que la récupération d'un ensemble de paramètres et le mode de stimulation neuromusculaire ont été exercés. L'architecture initiale du stimulateur prévoyait laisser la gestion de la stimulation temps réel au générateur de stimuli, mais comme certaines erreurs se sont glissées dans la réalisation du circuit intégré, la génération de la valeur numérique de l'amplitude en fonction du temps est réalisée par le microcontrôleur à des fins de test. L'amplitude ainsi que le sens du courant sont ensuite envoyés à l'étage analogique de sortie pour la génération du courant bidirectionnel.

5.6 Conclusion

Le prototype du stimulateur externe présenté dans ce chapitre utilise un clavier étendu ainsi que des menus structurés augmentant significativement la convivialité de l'interface usager. De cette façon, un utilisateur ayant un minimum d'expérience peut exploiter l'ensemble des fonctions disponibles. Les quatre espaces mémoires

ajoutent également un atout important en permettant de conserver les ensembles de paramètres les plus utilisés. Également, la réutilisation du circuit intégré permet de réduire la complexité du design de la partie analogique pouvant mener à un système de taille réduite. Enfin, l'architecture utilisée permet l'intégration facile de nouvelles fonctions sur les ports non-utilisés du microcontrôleur, tel que l'ajout d'un module de communication pouvant transformer le système en un contrôleur externe pour les stimulateurs implantables existants.

DISCUSSION GÉNÉRALE ET CONCLUSION

Parmi les sites de stimulation ayant fait l'objet d'études visant la récupération des fonctions vésicales chez les paraplégiques, les racines sacrées donnent les résultats les plus prometteurs. Cependant, la stimulation basse fréquence des nerfs sacrés mène à une contraction simultanée du détrusor et du sphincter urétral externe, perturbant la miction. La vidange poststimulus, le blocage anodique et la fatigue du sphincter sont autant de techniques de stimulation permettant de favoriser la miction. La stimulation sélective à l'aide de deux trains d'impulsions superposés détient plusieurs avantages sur les autres techniques.

Le sommaire des différents stimulateurs pour récupérer les fonctions urinaires montre les lacunes des systèmes présentement offerts. Au niveau commercial, il n'y a qu'un nombre très limité de stimulateurs disponibles. Le «Finnetech-Brindley Bladder System», le stimulateur proposé par Medtronic et le «Urogenital Stimulator» de Avery Laboratories, Inc. sont tous destinés à la vidange post-stimulus de la vessie. Au cours des années, plusieurs stimulateurs neuromusculaires dédiés à la récupération de la miction par fatigue du sphincter ou à la stimulation sélective ont été élaborés par l'équipe PolySTIM. Le point de départ de ces stimulateurs est un système multicanaux construit à partir d'un processeur dédié, introduit par Sawan. Les versions réalisées par Robin et Schneider, permettant la stimulation sélective, sont réalisées à partir de composants commerciaux montés en surface. La taille de

ces stimulateurs implantables restent relativement importante et ne permet pas le test *in vivo* chez les petits animaux tel le lapin ou le rat.

Les résultats d'expérimentation en phase chronique sur 8 chiens présentés dans l'article «Implantable Selective Stimulator to Improve Bladder Voiding: Design and Chronic Experiments in Dogs» publié dans «IEEE Transactions On Rehabilitation Engineering» montrent que la stimulation sélective réduit de plus de 50% le volume d'urine résiduel lorsque comparée avec une stimulation basse fréquence. Une étude exhaustive comparant la stimulation sélective aux autres techniques de stimulation mentionnées au chapitre 1 n'a pu être réalisée avec la littérature disponible. Les essais expérimentaux nécessaires à une telle comparaison dépassent largement le cadre d'un projet de maîtrise, sans parler des coûts associés à une étude de cette envergure.

L'implant présenté dans ce mémoire est le dernier d'une série de stimulateurs neuromusculaires. Il est basé sur un circuit intégré qui comporte un module numérique dédié à la génération de l'amplitude numérique de l'onde de stimulation sélective et une source de courant commandée. Les performances du circuit restent modestes lorsque comparées aux designs précédents à l'aide du facteur de qualité défini par St-Amand, lequel donne une place importante à la taille du circuit. La linéarité du convertisseur est toutefois comparable aux designs proposés antérieurement. Les besoins du système sont toutefois atteints pour une première fois dans une technologie utilisant une alimentation de 3.3 volts.

Enfin, le prototype de stimulateur externe basé sur le microcontrôleur E0C88316 d'Epson intègre une interface usager élaborée comprenant un clavier de 13 touches et un écran de 4 lignes de 16 caractères. À l'aide de cette interface, il est possible de modifier tous les paramètres de stimulation et de sauvegarder un ensemble de paramètres pour utilisation ultérieure. L'étage de stimulation utilise le circuit intégré dédié au stimulateur implantable, uniformisant de cette façon les étages de sortie des systèmes de stimulation utilisés à chacune des phases de la validation de la méthode de stimulation.

La suite naturelle du développement du stimulateur implantable consiste à intégrer les parties de décodage Manchester et de réception des données du système ainsi que la partie de régulation de l'alimentation. Aussi, l'intégration de nouvelles fonctions est de mise. Deux fonctions déjà incluses dans les prototypes développés à l'aide de composants discrets permettent de passer en mode de stimulation permanente pour contrer l'incontinence et offrent la mesure de la résistance du contact nerf-électrode, utilisée en tant qu'outil de diagnostique. La mesure de la résistance nécessite toutefois un lien de communication duplex pour retourner l'information obtenue vers le contrôleur.

Du côté du stimulateur externe, l'architecture présentée au chapitre 5 permet l'intégration de nouvelles fonctions au niveau logiciel, sans modifications majeures de la partie matérielle. L'éventuelle réalisation d'un circuit imprimé doit permettre l'ajout de cartes d'expansion pour, entre autres, un module de communication

duplex transformant le stimulateur externe en contrôleur d'implant. La qualité de l'interface usager et l'architecture existante du stimulateur externe ferait d'un éventuel système, un contrôleur polyvalent pouvant être utilisé avec différents types d'implant présentement utilisés lors des essais en phase chronique.

L'une des constatations remarquées pendant la réalisation du microstimulateur est l'importance d'une méthodologie complète et éprouvée. Dans un premier temps, au niveau de la vérification de la partie numérique. Il est essentiel de rédiger des spécifications claires et pour tous les modules et de joindre le modèle fonctionnel de chacun des blocs analogiques. De cette façon, la simulation de la partie numérique et des modèles fonctionnels permet d'identifier les problèmes d'interface entre les différents modules avant la fabrication. Il est aussi essentiel de simuler de façon approfondie tous les modes de fonctionnement de la partie numérique et d'éprouver les limites pour chacun des éléments du design. Si le système atteint une complexité critique, il peut également être nécessaire de définir un modèle de référence de l'ensemble du système pour comparer les résultats obtenus par les descriptions synthétisables avec ceux escomptés.

En second lieu, l'analyse des résultats de synthèse et du placement et routage doit être systématique et complète. Elle doit comprendre l'analyse des délais de propagation, des fréquences maximales d'opération, de la distribution des horloges et des signaux passant d'un domaine d'horloge à un autre, s'il y a lieu. Comme la méthodologie utilisée pour la réalisation de circuits intégrés comporte un grand

nombre d'étapes, il est recommandé de la vérifier avec un design de complexité réduite.

Aussi, la méthodologie utilisée devrait avoir recours à des scripts plutôt qu'à l'interface graphique des outils employés. L'utilisation de fichiers de commandes permet d'archiver ces fichiers à l'aide d'outils de contrôle des versions et de reproduire sans équivoque une séquence d'opérations. Les fichiers de commandes documentent la méthodologie et évitent ainsi les incertitudes sur la valeur finale des différents paramètres utilisés pour la synthèse, le placement et le routage, et les étapes de vérification.

BIBLIOGRAPHIE

- [1] —. *InterStim®therapy - Physician and Hospital Staff Manual*. Minneapolis, USA, 2001.
- [2] N. Accornero, G. Bini, G. L. Lenzi, and M. Manfredi. Selective activation of peripheral nerve fiber groups of different diameter by triangular shaped stimulus pulses. *J. Physiol.*, 273(3):539–560, 1977.
- [3] K. Arabi. Conception d'un microstimulateur neuromusculaire destiné à la récupération des dysfonctions urinaires. Master's thesis, École Polytechnique de Montréal, 1994.
- [4] K. Arabi and M. Sawan. Electronic design and realization of a new multi-programmable microimplant for neuromuscular electrical stimulation. *IEEE Trans. Rehab. Eng.*, 7(2):204–214, 1999.
- [5] E. Blair and J. Erlanger. A comparison of the characteristics of axons through their individual electrical responses. *Amer. J. Physiol.*, 106:524–564, 1933.
- [6] S. Bourret. Proposition d'une stratégie de stimulation uniarticulaire et mise au point d'une source de courant dédiée à la stimulation neuromusculaire. Master's thesis, École Polytechnique de Montréal, 1999.

- [7] W. H. Boyce, J. E. Lathem, and L. D. Hunt. Research related to the development of an artificial electrical stimulator for the paralyzed human bladder: A review. *J. Urol.*, 91:41–51, 1964.
- [8] S. Boyer, M. Abdel-Gawad, T. M. Abdel-Baky, M. Sawan, and M. M. Elhilali. Selective neural stimulation to improve bladder voiding: Chronic experiments in dogs. In *Proc. IFESS Conf.*, Lucerne, 1998.
- [9] S. Boyer, M. Sawan, M. Abdel-Gawad, S. Robin, and M. M. Elhilali. Implantable selective stimulator to improve bladder voiding: Design and chronic experiments in dogs. *IEEE Trans. Rehab. Eng.*, 8:464–470, 2000.
- [10] G. S. Brindley. An implant to empty the bladder or close the urethra. *J. Neurology*, 40:358–369, 1977.
- [11] G. S. Brindley. The first 500 patients with sacral anterior root stimulator implants: General description. *Paraplegia*, 32:795–805, 1994.
- [12] Encyclopaedia Britannica, editor. *Encyclopaedia Britannica*, volume 7, pages 859–860. Chicago, 1984.
- [13] S. B. Brummer, L. S. Robblee, and F. T. Hambrecht. Criteria for selecting electrodes for electrical stimulation: Theoretical and practical considerations. *Annals New York Academy of Sciences*, pages 159–171, 1983.

- [14] J.-M. Buzelin and M. Averous. *Urodynamique: bas appareil urinaire*. Masson, Paris; New York; Barcelone, 1984.
- [15] Société canadienne de micro électronique. site Web. <http://www.cmc.ca/>.
- [16] R. R. Carter, D. B. McCreery, B. J. Woodford, L. A. Bullara, and W. F. Agnew. Micturition control by microstimulation of the sacral spinal cord of the cat: Acute studies. *IEEE Trans. Rehab. Eng.*, 3:206–214, 1995.
- [17] M.-A. Crampon. Conception et réalisation d'électrodes neuronales dédiées à des stimulateurs électroniques implantables. Master's thesis, École Polytechnique de Montréal, 1999.
- [18] PolySTIM Laboratoire de recherche en neurotechnologies. site Web. <http://www.polystim.polymtl.ca/>.
- [19] A. Djemouai, P. Vaillancourt, M. Sawan, and M. Slamani. Performance optimization of a radio-frequency coupling technique. In *Proc. IFESS Conf.*, Vancouver, 1997.
- [20] Z. P. Fang and J. T. Mortimer. Selective activation of small motor axons by quasitrapezoidal current pulses. *IEEE Trans. Biomed. Eng.*, 38(2):168–174, 1991.

- [21] H. Friedman, B. S. Nashold, and P. Senechal. Spinal cord stimulation and bladder function in normal and paraplegic animals. *J. Neurosurg.*, 36:430–437, 1972.
- [22] J. H. Grimes, B. S. Nashold, and D. P. Currie. Chronic electrical stimulation of the paraplegic bladder. *J. Urol.*, 109:242–245, 1973.
- [23] The Dobelle Group. site Web, 2002. <http://www.dobelle.com/>.
- [24] M. Hassouna, J.S. Li, and M. M. Elhilali. Dog as animal model for neurostimulation. *Neurourol. Urodynam.*, 13:159–167, 1994.
- [25] M. Haugland. A flexible method for fabrication of nerve cuff electrodes. In *IEEE EMBS Conf.*, Amsterdam, 1996.
- [26] B. Holmquist and W. J. Staubitz. The role of the pudendal nerve in connection with electronic emptying of the neurogenic cord bladder in dogs. *J. Urol.*, 98:198–204, 1967.
- [27] U. Jonas, J. P. Heine, and E. A. Tanagho. Studies on the feasibility of urinary bladder evacuation by direct spinal cord stimulation. *Invest. Urol.*, 13:142–153, 1975.
- [28] J. S. Li, M. Hassouna, M. Sawan, F. Duval, and M. M. Elhilali. Electrical stimulation induced sphincter fatigue during voiding. *J. Urol.*, 148:949–952, 1992.

- [29] J. S. Li, M. Hassouna, M. Sawan, F. Duval, and M. M. Elhilali. Long-term effect of sphincteric fatigue during bladder neurostimulation. *J. Urol.*, 153:238–242, 1995.
- [30] D. R. McNeal. *2000 Years of Electrical Stimulation*. Dekker, New York, f.t. hambrecht & j.b. reswick edition, 1977.
- [31] S. Quérin and L. Valiquette. *Physiopathologie des maladies du rein et des voies urinaires*. Edisem Inc., 2000.
- [32] N. J. M. Rijkhoff, H. Wijkstra, P. E. V. van Kerrebroeck, and F. M. J. Debruyne. Urinary bladder control by electrical stimulation: Review of electrical stimulation techniques in spinal cord injury. *Neurourol. Urodynam.*, 16:39–53, 1997.
- [33] S. Robin. Réalisation et tests d’un système implantable dédié à la stimulation neurale sélective. Master’s thesis, École Polytechnique de Montréal, 1998.
- [34] S. Robin, M. Sawan, M. Abdel-Gawad, T. M. Abdel-Baky, and M. M. Elhilali. Implantable stimulation system dedicated for neural selective stimulation. *Med. Bio. Eng. Comp.*, pages 490–492, 1998.
- [35] M. Sawan. *Conception, réalisation et tests in vivo des stimulateurs neuromusculaires destinés aux patients souffrant de dysfonctions urinaires*. PhD thesis, Université de Sherbrooke, 1990.

- [36] M. Sawan, F. Duval, M. Hassouna, and M. M. Elhilali. A new transcutaneous fully-programmable neural stimulator. *Int. J. Microcomputer App.*, 13(3):142–147, 1994.
- [37] M. Sawan, F. Duval, M. Hassouna, J. S. Li, and M. M. Elhilali. A new bladder stimulator - hand-held controller and miniaturized implant: Preliminary results on dogs. *Biomed. Instrum. Tech.*, 27:143–149, 1993.
- [38] M. Sawan, F. Duval, M. Hassouna, J. S. Li, M. M. Elhilali, J. Lachance, Marc Leclair, Soheyl Pourmehdi, and Jaouhar Mouïne. Computerized transcutaneous control of a multichannel implantable urinary prosthesis. *IEEE Trans. Biomed. Eng.*, 39:600–609, 1992.
- [39] M. Sawan, M. Hassouna, J. S. Li, F. Duval, and M. M. Elhilali. Stimulator design and subsequent stimulation parameter optimization for controlling micturition and reducing urethral resistance. *IEEE Trans. Rehab. Eng.*, 4(1):39–46, 1996.
- [40] E. Schneider. Conception et évaluation d'un système de stimulation électrique neurale dédié à la réhabilitation des fonctions vésicales. Master's thesis, École Polytechnique de Montréal, 2001.
- [41] H. S. Shaker, L. M. Tu, S. Robin, K. Arabi, M. Hassouna, M. Sawan, and M. M. Elhilali. Reduction of bladder outlet resistance by selective sacral root

- stimulation using high-frequency blockade in dogs: An acute study. *J. Urol.*, 160:901–907, 1998.
- [42] R. St-Amand. Conception d’une source de courant servant d’étage de sortie à un stimulateur neuromusculaire. Master’s thesis, École Polytechnique de Montréal, 1995.
- [43] R. St-Amand, M. Sawan, and Y. Savaria. Design and optimisation of a low dc offset current source dedicated to implantable miniaturized stimulators. *Analog Integrated Circuits and Signal Processing*, 11:47–71, 1996.
- [44] S. L. Stover, J. A. Delisa, and G. G. Whiteneck. *Spinal Cord Injury: Clinical Outcomes from the Model Systems*. Aspen Publishers, Inc., 1995.
- [45] J. G. Susset and Z. N. Boctor. Electrical stimulation of the bladder: An experimental study. *Invest. Urol.*, 5(20):20–29, 1967.
- [46] A. Talalla, J. W. Bloom, and N. Quang. Fes for bladder: Direct or indirect means? *Pace*, 10:240–245, 1987.
- [47] J. S. Walter, R. Sidarous, C. J. Robinson, J. S. Wheeler, and R. D. Wurster. Comparison of direct bladder and sacral nerve stimulation in spinal cats. *J. Rehab. Research & Development*, 29(2):13–22, 1992.
- [48] J. S. Walter, J. S. Wheeler, Jr., G. Creasey, R. Chintam, L. Riedy, K. Bruninga, E. Collins, B. Nemchausky, and D. Anderson. Optimization of

sacral ventral root stimulation following sci: Two case reports with six-month follow-up. *J. Spinal Cord Med.*, 21:211–219, 1998.

- [49] J. S. Walter, J. S. Wheller, and R. B. Dunn. Dynamic bulbocavernosus reflex: Dyssynergia evaluation following sci. *J. Am. Paraplegia Soc.*, 17:140–145, 1994.
- [50] J. S. Walter, J. S. Wheller, C. J. Robinson, and R. D. Wurster. Inhibiting the hyperreflexic bladder with electrical stimulation in a spinal animal model. *Neurourol. Urodynam.*, 12:241–253, 1993.
- [51] J. Yue and G. Ahuja, editors. *Toronto Notes: MCCQE 2001 Review Notes*. MCCQE Notes Ltd., seventeenth edition, 2001.

Annexe I

Code VHDL du stimulateur implantable et de son environnement de vérification

I.1 Code VHDL du «test bench»

```
-----  
-- Ecole Polytechnique de Montreal  
-- Groupe de Recherche en Microelectronique  
-- Equipe de Recherche en Neurotechnologies  
--  
-- Realisation d'un micro-stimulateur implantable totalement integre  
--  
-- Auteur: Stephane Boyer  
--  
-- Testbench de "stimulateur"  
--  
-- Fichier: stimulateur_tb.vhd  
-----  
-- Date Modification  
--  
-- 15 nov.98 Ajout du module header_detect  
-- 21 nov.98 Ajout du module source_ctrl  
-- 24 nov.98 Ajout de port sur stimulateur, retrait de source_ctrl  
-----
```

```
LIBRARY ieee;  
USE ieee.STD_LOGIC_1164.ALL;  
USE ieee.STD_LOGIC_signed.ALL;  
USE ieee.STD_LOGIC_arith.ALL;  
USE WORK.header_def.ALL;  
  
ENTITY stimulateur_tb IS  
END stimulateur_tb;  
  
ARCHITECTURE test OF stimulateur_tb IS  
  COMPONENT stimulateur  
    PORT(clk, por,  
          shift_en, shift_in : IN STD_LOGIC;  
          shift_out          : OUT STD_LOGIC;  
          test, go           : IN STD_LOGIC;  
          no_detect,  
          stim_detect,  
          test_detect        : OUT STD_LOGIC;  
          stim_n1,  
          stim_n2,  
          stim_p1,  
          stim_p2            : OUT STD_LOGIC;  
          amp                 : OUT STD_LOGIC_VECTOR(4 DOWNTO 0);
```

```

        plage                : OUT STD_LOGIC_VECTOR(2 DOWNTO 0));
END COMPONENT;

SIGNAL clk, por,
        shift_en, shift_in,
        shift_out,
        test, go            : STD_LOGIC;
SIGNAL no_detect,
        stim_detect,
        test_detect        : STD_LOGIC;
SIGNAL source0             : STD_LOGIC_VECTOR(1 TO 4);
-- n1,n2,p1,p2: inactif = 3, sens1 = 9, sens2 = 6
SIGNAL amp                 : STD_LOGIC_VECTOR(4 DOWNTO 0);
SIGNAL plage               : STD_LOGIC_VECTOR(2 DOWNTO 0);

BEGIN
module_stimulateur : stimulateur PORT MAP(
    clk => clk, por => por,
    shift_en => shift_en,
    shift_in => shift_in,
    shift_out => shift_out,
    test => test, go => go,
    no_detect => no_detect,
    stim_detect => stim_detect,
    test_detect => test_detect,
    stim_n1 => source0(1),
    stim_n2 => source0(2),
    stim_p1 => source0(3),
    stim_p2 => source0(4),
    amp => amp,
    plage => plage);

horloge : PROCESS
BEGIN
    clk <= '1', '0' AFTER 5 ns;
    WAIT FOR 10 ns;
END PROCESS Horloge;

envoi_donnees : PROCESS
VARIABLE stream      : STD_LOGIC_VECTOR(86 DOWNTO 0);
BEGIN
    WAIT UNTIL clk'EVENT AND clk='0';

    -- initialisation des donnees pour le test
    stream(86 DOWNTO 77) := stim_header;          -- entete 10 bits
    stream(76 DOWNTO 72) := "00000";             -- hfa   5 bits h"00"
    stream(71 DOWNTO 56) := "0000000000000000";  -- hfp  16 bits h"0000"
    stream(55 DOWNTO 48) := "00000000";          -- hfi   8 bits h"00"
    stream(47 DOWNTO 40) := "00000000";          -- hfw   8 bits h"00"
    stream(39 DOWNTO 35) := "00101";              -- lfa   5 bits h"05"
    stream(34 DOWNTO 19) := "0000000000010000";  -- lfp  16 bits h"0010"
    stream(18 DOWNTO 11) := "00000000";          -- lfi   8 bits h"00"
    stream(10 DOWNTO 3)  := "00000100";          -- lfw   8 bits h"04"
    stream(2 DOWNTO 0)   := "001";               -- plage 3 bits h"1"
                                           -- TOTAL 87 bits

    shift_in <= '0';

    -- decoil pour le debut de la sequence
    por <= '0', '1' AFTER 100 ns; -- actif bas
    WAIT FOR 100 ns;

```

```

-----
-- envoi de la premiere sequence de stimulation
FOR index_1 IN 86 DOWNT0 0 LOOP
    shift_in <= stream(index_1);
    WAIT UNTIL clk'EVENT AND clk='0';
END LOOP;
shift_in <= 'X';
WAIT UNTIL clk'EVENT AND clk='0';
shift_in <= '1';
WAIT UNTIL clk'EVENT AND clk='0';

-----

-- attendre pour verifier les resultats
WAIT FOR 1000 ns;

-----

-- initialisation des donnees pour le test (2)
stream(86 DOWNT0 77) := stim_header;
stream(76 DOWNT0 72) := "00010";           --hfa 5 bits
stream(71 DOWNT0 56) := "0000000000000100"; --hfp 16 bits
stream(55 DOWNT0 48) := "00000001";        --hfi 8 bits
stream(47 DOWNT0 40) := "00000001";        --hfw 8 bits
stream(39 DOWNT0 35) := "00011";           --lfa 5 bits
stream(34 DOWNT0 19) := "0000000000010000"; --lfp 16 bits
stream(18 DOWNT0 11) := "00000000";        --lfi 8 bits
stream(10 DOWNT0 3)  := "00000100";        --lfw 8 bits
stream(2 DOWNT0 0)   := "010";             --plage 3 bits h"2"

-----

-- envoi de la seconde sequence de stimulation

por <= '0', '1' AFTER 100 ns; -- actif bas
WAIT FOR 100 ns;
FOR index_2 IN 86 DOWNT0 0 LOOP
    shift_in <= stream(index_2);
    WAIT UNTIL clk'EVENT AND clk='0';
END LOOP;

-----

-- attendre pour verifier les resultats
WAIT FOR 1000 ns;

-----

-- mode test: generation de la rampe
por <= '0', '1' AFTER 100 ns; -- actif bas
WAIT FOR 100 ns;
stream(86 DOWNT0 77) := test_header;
stream(76 DOWNT0 0)  := (OTHERS => '0');
FOR index_3 IN 86 DOWNT0 0 LOOP
    shift_in <= stream(index_3);
    WAIT UNTIL clk'EVENT AND clk='0';
END LOOP;
WAIT FOR 500 ns;

END PROCESS envoi_donnees;

boucle : PROCESS (no_detect, stim_detect, test_detect)
BEGIN
    shift_en <= no_detect;
    go <= stim_detect;
    test <= test_detect;
END PROCESS;

```

```

END test;

CONFIGURATION stimulateur_tb_cfg OF stimulateur_tb IS
  FOR test
  END FOR;
END stimulateur_tb_cfg;

```

I.2 Code VHDL du stimulateur implantable

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur: Stephane Boyer
--
-- Module: Jonction de tous les modules
--   Pour la fabrication du CI du 2 decembre 1998
-- (niveau 3)
--
-- Fichier: stimulateur.vhd
-----
-- Date Modification
--
-- 15 nov.98  Ajout du module head_detect
-- 21 nov.98  Ajout de la sortie plage pour la source
-- 21 nov.98  Retrait de la sortie shift_out du shift_reg
-- 24 nov.98  Ajout du module source_ctrl
-- 25 nov.98  Ajout des ports intermediaires "sign" et "en_output"
-----

```

```

LIBRARY ieee;
USE ieee.STD_LOGIC_1164.ALL;
USE ieee.STD_LOGIC_signed.ALL;
USE ieee.STD_LOGIC_arith.ALL;

ENTITY stimulateur IS
  PORT(clk, por,
        shift_en, shift_in  : IN  STD_LOGIC;
        shift_out           : OUT STD_LOGIC;
        test, go            : IN  STD_LOGIC;
        no_detect,
        stim_detect,
        test_detect         : OUT STD_LOGIC;
        stim_n1, stim_n2,
        stim_p1, stim_p2    : OUT STD_LOGIC;
        amp                 : OUT STD_LOGIC_VECTOR(4 DOWNTO 0);
        plage               : OUT STD_LOGIC_VECTOR(2 DOWNTO 0));
END stimulateur;

```

```

ARCHITECTURE struct OF stimulateur IS

  COMPONENT head_detect

```

```

    PORT(clk, por,
          shift_en,
          shift_in      : IN STD_LOGIC;
          shift_out,
          no_detect,
          stim_detect,
          test_detect   : OUT STD_LOGIC);
END COMPONENT;

```

```

COMPONENT shift_reg
  PORT(clk, por,
        shift_in,
        shift_en      : IN STD_LOGIC;
        hfa, lfa       : OUT STD_LOGIC_VECTOR(4 DOWNTO 0);
        hfp, lfp       : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
        hfw, lfw,
        hfi, lfi       : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        plage          : OUT STD_LOGIC_VECTOR(2 DOWNTO 0));
END COMPONENT;

```

```

COMPONENT uc
  PORT(clk, go,
        test          : IN STD_LOGIC;
        hfa, lfa       : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
        hfp, lfp       : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
        hfi, lfi,
        hfw, lfw       : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        en, sign       : OUT STD_LOGIC;
        amp            : OUT STD_LOGIC_VECTOR(4 DOWNTO 0));
END COMPONENT;

```

```

COMPONENT source_ctrl
  PORT(source_en,
        source_sign   : IN STD_LOGIC;
        stim_n1,
        stim_n2,
        stim_p1,
        stim_p2       : OUT STD_LOGIC);
END COMPONENT;

```

```

SIGNAL shfa, slfa : STD_LOGIC_VECTOR(4 DOWNTO 0);
SIGNAL shfp, slfp : STD_LOGIC_VECTOR(15 DOWNTO 0);
SIGNAL shfw, slfw : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL shfi, slfi : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL shift_tmp  : STD_LOGIC;
SIGNAL en_output, sign : STD_LOGIC;

```

BEGIN

```

  module_head_detect : head_detect PORT MAP(
    clk => clk, por => por,
    shift_en => shift_en,
    shift_in => shift_tmp,
    shift_out => shift_out,
    no_detect => no_detect,
    stim_detect => stim_detect,
    test_detect => test_detect);

```

```

  module_shift_reg : shift_reg PORT MAP(
    clk => clk, por => por,
    shift_in => shift_in,
    shift_en => shift_en,
    hfa => shfa, lfa => slfa,

```

```

    hfp => shfp, lfp => slfp,
    hfi => shfi, lfi => slfi,
    hfw => shfw, lfw => slfw,
    plage => plage);

module_uc : uc PORT MAP(
    clk => clk, go => go, test => test,
    hfa => shfa, lfa => slfa,
    hfp => shfp, lfp => slfp,
    hfi => shfi, lfi => slfi,
    hfw => shfw, lfw => slfw,
    en => en_output,
    sign => sign,
    amp => amp);

module_source_ctrl : source_ctrl PORT MAP(
    source_en => en_output,
    source_sign => sign,
    stim_n1 => stim_n1,
    stim_n2 => stim_n2,
    stim_p1 => stim_p1,
    stim_p2 => stim_p2);

-- shift_out de shift_reg: hfa(4)
shift_tmp <= shfa(4);

END struct;

```

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur: Stephane Boyer
--
-- Module: Registre de reception des parametres
--   Registre a decalage qui recoit les parametres de
--   stimulation. Module temporaire pour la fabrication
--   du CI (run du 2 decembre 1998).
--   (niveau 2)
--
-- Fichier: shift_reg.vhd
-----
-- Date Modification
--
-- 18 nov.98  Ajout d'un bit dans stream pour separer shift_out de hfa(4)
-- 21 nov.98  Retrait de la sortie shift_out, remplacer par hfa(4)
-- 21 nov.98  Ajout de 3 bits pour les plages de la source
-- 24 nov.98  Entree 'por' -> synchrone
-----

```

```

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

```

```

ENTITY shift_reg IS
  PORT(clk, por,
        shift_en,
        shift_in  : IN STD_LOGIC;
        hfa, lfa  : OUT STD_LOGIC_VECTOR(4 DOWNTO 0);
        hfp, lfp  : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
        hfi, lfi,
        hfw, lfw  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        plage     : OUT STD_LOGIC_VECTOR(2 DOWNTO 0));
END shift_reg;

```

```

ARCHITECTURE behav OF shift_reg IS
  SIGNAL stream : STD_LOGIC_VECTOR(76 DOWNTO 0);
BEGIN
  balayage : PROCESS
  BEGIN
    WAIT UNTIL clk'EVENT AND clk='1';
    IF por ='0' THEN -- actif bas
      stream(76 DOWNTO 0) <= (OTHERS => '0');
    ELSIF shift_en = '1' THEN
      stream(76 DOWNTO 1) <= stream(75 DOWNTO 0);
      stream(0) <= shift_in;
    END IF;
  END PROCESS balayage;

```

```

  assignation_sorties : PROCESS (stream)
  BEGIN
    hfa <= stream(76 DOWNTO 72); -- 5 bits -- shift_out <= hfa(4)
    hfp <= stream(71 DOWNTO 56); -- 16 bits
    hfi <= stream(55 DOWNTO 48); -- 8 bits
  END PROCESS assignation_sorties;

```

```
hfw  <= stream(47 DOWNT0 40);  -- 8 bits
lfa  <= stream(39 DOWNT0 35);  -- 5 bits
lfp  <= stream(34 DOWNT0 19);  -- 16 bits
lfi  <= stream(18 DOWNT0 11);  -- 8 bits
lfw  <= stream(10 DOWNT0 3);   -- 8 bits
plage <= stream(2 DOWNT0 0);   -- 3 bits
END PROCESS assignation_sorties;
END behav;
```

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur: Stephane Boyer
--
-- Module: Registre de detection d'entete
--   Registre a decalage compare les donnees entrantes a
--   l'entete de donnees ou de test. Ce module verouille
--   lorsque l'entete est trouvee. Module temporaire pour
--   la fabrication du CI (run du 2 decembre 1998).
--   (niveau 2)
--
-- Fichier: head_detect.vhd
-----

-- Date Modification
--
-- 24 nov.98   Entree 'por' -> synchrone
-----

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

PACKAGE header_def IS  -- declaration des entetes
  CONSTANT test_header : STD_LOGIC_VECTOR(9 DOWNTO 0) := "0111111100";
  CONSTANT stim_header : STD_LOGIC_VECTOR(9 DOWNTO 0) := "0111111101";
END header_def;

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE ieee.std_logic_arith.ALL;
USE WORK.header_def.ALL;

ENTITY head_detect IS
  PORT(clk, por,
        shift_en,
        shift_in      : IN STD_LOGIC;
        shift_out,
        no_detect,
        stim_detect,
        test_detect    : OUT STD_LOGIC);
END head_detect;

ARCHITECTURE behav OF head_detect IS

  SIGNAL stream : STD_LOGIC_VECTOR(9 DOWNTO 0);

BEGIN
  balayage : PROCESS
  BEGIN
    WAIT UNTIL clk'EVENT AND clk='1';
    IF por='0' THEN  -- actif bas
      stream(9 DOWNTO 0) <= (OTHERS => '0');
    ELSIF shift_en = '1' THEN
      stream(9 DOWNTO 1) <= stream(8 DOWNTO 0);
      stream(0) <= shift_in;
    END IF;
  END PROCESS;

```

```
END IF;
END PROCESS balayage;

assignation_sorties : PROCESS (stream)
BEGIN
    shift_out <= stream(9);
    IF stream = stim_header THEN
        no_detect <= '0';
        stim_detect <= '1';
        test_detect <= '0';
    ELSIF stream = test_header THEN
        no_detect <= '0';
        stim_detect <= '0';
        test_detect <= '1';
    ELSE
        no_detect <= '1';
        stim_detect <= '0';
        test_detect <= '0';
    END IF;
END PROCESS assignation_sorties;
END behav;
```

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur:  Pierre Vaillancourt
-- Revision: Stephane Boyer
--
-- Module:  Generation de la forme d'onde
--          Ce module comprend les modules synchro, dsp et somme.
--          (niveau 2)
--
-- Fichier: uc.vhd
-----

-- Date Modification
--
--
-----

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE ieee.std_logic_arith.ALL;

ENTITY uc IS
    PORT(clk,                                -- horloge
          go,                                -- generer les impulsions
          test : IN STD_LOGIC;              -- generer une rampe
          hfa, lfa : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
          hfp, lfp : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
          hfi, lfi,
          hfw, lfw : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
          en,
          sign : OUT STD_LOGIC;              -- enable output
          amp : OUT STD_LOGIC_VECTOR(4 DOWNTO 0)); -- signe de l'impulsion
-- amplitude
END uc;

ARCHITECTURE struct OF uc IS
    SIGNAL sync_l, en_l, sign_l, sync_h, en_h, sign_h : STD_LOGIC;
    SIGNAL amp_l, amp_h : STD_LOGIC_VECTOR(4 DOWNTO 0);

    COMPONENT synchro
        PORT(clk, go : IN STD_LOGIC;
              fp : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
              sync : OUT STD_LOGIC);
    END COMPONENT;

    COMPONENT dsp
        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 COMPONENT;

    COMPONENT somme
        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 COMPONENT;

```

```

END COMPONENT;

BEGIN
  synchro_l : synchro PORT MAP(
    clk => clk, go => go,
    fp => lfp, sync => sync_l);

  synchro_h : synchro PORT MAP(
    clk => clk, go => go,
    fp => hfp, sync => sync_h);

  dsp_l : dsp PORT MAP(
    clk => clk, go => go,
    sync => sync_l, fw => lfw, fi => lfi, fa => lfa,
    en => en_l, sign => sign_l, amp => amp_l);

  dsp_h : dsp PORT MAP(
    clk => clk, go => go,
    sync => sync_h, fw => hfw, fi => hfi, fa => hfa,
    en => en_h, sign => sign_h, amp => amp_h);

  la_somme : somme PORT MAP(
    clk => clk,
    en_l => en_l, sign_l => sign_l,
    en_h => en_h, sign_h => sign_h,
    test => test, go => go,
    amp_l => amp_l, amp_h => amp_h,
    en_somme => en,
    sign_somme => sign,
    amp_somme => amp);

END struct;

```

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- PolyStim - Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur: Alexandre Beauchamps-Parent
-- Revision: Stephane Boyer
--
-- Module generant l'amplitude et le signe d'un train d'impulsions
-- en fonction de la largeur d'impulsion et de l'interimpulsion
-- (niveau 1)
--
-- Fichier: dsp.vhdl
-----

-- Date      Modification
--
-----

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;

  -- Horloge a 300 kHz
  -- Genere les sorties
  -- Debut de cycle
  -- Largeur d'impulsion
  -- Interimpulsion
  -- Amplitude
  -- Enable output (sortie)
  -- Signe du pulse (sortie)
  -- Amplitude (sortie)

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";
      -- arret
    ELSE
      -- produire les sorties
      IF sync='1' THEN
        compteur2 := "00000000";
        IF compteur2=fw THEN
          etat := "00";
        ELSE
          etat := "01";
        end IF;
      end IF;
    end IF;
  end PROCESS

```

```

END IF;
case etat is
  WHEN "00" => en <='0';          -- etat d'attente
                sign <='-';
                amp <="00000";
  WHEN "01" => en <= '1';        -- etat pulse positif
                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';        -- etat interpulse
                sign <= '-';
                amp <= "00000";
                compteur3 := compteur3+'1';
                IF compteur3= fi THEN
                  etat := "11";
                END IF;
  WHEN "11" => en <= '1';        -- etat pulse negatif
                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;

```

```

-----
-- 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
--
-- Auteur: Alexandre Beauchamp Parent
-- Revision: Stephane Boyer
--
-- Module: Synchronisation du debut du signal a generer avec
--         tous les FP coups d'horloge.
--         (niveau 1)
--
-- Fichier: synchro.vhd
-----
-- 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 DOWNT0 0);
        sync    : OUT STD_LOGIC);
END synchro;

```

```

ARCHITECTURE algo OF synchro IS
BEGIN
  PROCESS
    VARIABLE compteur1 : STD_LOGIC_VECTOR(15 DOWNT0 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;

```

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur: Alexandre Beauchamp-Parent
-- Revision: Stephane Boyer
--
-- Module: Somme les amplitudes et genere une rampe en mode test.
--          (niveau 1)
--
-- Fichier: somme.vhd
--

```

```

-----
-- Date Modification
--
-- 16 nov.98   Mode test: test = 1 (sans go = 1)
--

```

```

-----
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' AND test /= '1' THEN
      sign_rampe := '0';
      amp_rampe  := "00000";
    END IF;
    IF test='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 DOWNT0 0);
END IF;
    WHEN OTHERS => NULL;
END CASE;
    WHEN OTHERS => NULL;
END CASE;
END IF;
END PROCESS;
END algo;
```

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur:   Stephane Boyer
--
-- Module:   Module combinatoire controlant les transistors de
--           1'architecture en H pour deux sources commandees.
-- (niveau 3)
--
-- Fichier:  source_ctrl.vhd
-----
-- Date Modification
--
--
-----

```

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE ieee.std_logic_arith.ALL;

```

```

ENTITY source_ctrl IS
    PORT(source_en,
          source_sign : IN STD_LOGIC;
          stim_n1,
          stim_n2,
          stim_p1,
          stim_p2      : OUT STD_LOGIC);
END source_ctrl;

```

```

ARCHITECTURE combinatoire OF source_ctrl IS

```

```

BEGIN
    -- transistors p acitfs bas
    stim_n1 <= source_en AND NOT(source_sign);
    stim_p1 <= NOT(source_en AND NOT(source_sign));
    stim_n2 <= source_en AND source_sign;
    stim_p2 <= NOT(source_en AND source_sign);

END combinatoire;

```

```

-----
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- Realisation d'un micro-stimulateur implantable totalement integre
--
-- Auteur: Stephane Boyer
--
-- Module: Comprend les pads
--   Pour la fabrication du CI du 2 decembre 1998
-- (niveau 4)
--
-- Fichier: chip_pads.vhd
-----

```

```

-- Date Modification
--
-----

```

```

LIBRARY ieee;
USE ieee.STD_LOGIC_1164.ALL;

ENTITY source_s0 IS
  PORT(stim_n1,
        stim_n2,
        stim_p1,
        stim_p2      : IN      STD_LOGIC;
        D4, D3, D2,
        D1, D0        : IN      STD_LOGIC;
        vref           : INOUT STD_LOGIC;
        Iout_m,
        Iout_p         : INOUT STD_LOGIC;
        VDD, VSS       : INOUT STD_LOGIC);
END source_s0;

```

```

ARCHITECTURE dummy OF source_s0 IS
BEGIN
END dummy;

```

```

-----
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.ALL;

ENTITY source_s1 IS
  PORT(P2, P1, P0      : IN      STD_LOGIC;
        D4, D3, D2,
        D1, D0         : IN      STD_LOGIC;
        Iout           : INOUT STD_LOGIC;
        VDD, VSS       : INOUT STD_LOGIC);
END source_s1;

```

```

ARCHITECTURE dummy OF source_s1 IS
BEGIN
END dummy;

```

```

-----
LIBRARY ieee;
LIBRARY padsp35;
USE ieee.STD_LOGIC_1164.ALL;
USE PADSP35.ALL;

```

ENTITY chip_2dec IS

```

    PORT(wclk, wpor      : IN    STD_LOGIC;
          wshift_en      : INOUT STD_LOGIC;
          wshift_in      : IN    STD_LOGIC;
          wshift_out     : OUT   STD_LOGIC;
          wtest, wgo     : INOUT STD_LOGIC;
          wio_mode       : IN    STD_LOGIC;
          -- no scan_chain
          -- wtest_se     : IN    STD_LOGIC;
          -- wtest_si1    : IN    STD_LOGIC;
          -- wtest_si2    : IN    STD_LOGIC;
          wn1, wn2,
          wp1, wp2       : INOUT STD_LOGIC;
          wamp           : INOUT STD_LOGIC_VECTOR(4 DOWNTO 0);
          wvref          : IN    STD_LOGIC; -- analogic
          woutput_s0_a   : OUT   STD_LOGIC; -- analogic
          woutput_s0_b   : OUT   STD_LOGIC; -- analogic
          woutput_s1_a   : OUT   STD_LOGIC; -- analogic
END chip_2dec;
```

ARCHITECTURE struct OF chip_2dec IS

COMPONENT stimulateur

```

    PORT(clk, por,
          shift_en, shift_in : IN  STD_LOGIC;
          shift_out          : OUT STD_LOGIC;
          test, go           : IN  STD_LOGIC;
          no_detect,
          stim_detect,
          test_detect        : OUT STD_LOGIC;
          stim_n1, stim_n2,
          stim_p1, stim_p2   : OUT STD_LOGIC;
          amp                : OUT STD_LOGIC_VECTOR(4 DOWNTO 0);
          plage              : OUT STD_LOGIC_VECTOR(2 DOWNTO 0));
END COMPONENT;
```

COMPONENT source_s0

```

    PORT(stim_n1,
          stim_n2,
          stim_p1,
          stim_p2   : IN  STD_LOGIC;
          D4, D3, D2,
          D1, D0    : IN  STD_LOGIC;
          vref       : IN  STD_LOGIC;
          lout_m,
          lout_p     : OUT  STD_LOGIC;
          VDD, VSS   : INOUT STD_LOGIC);
END COMPONENT;
```

COMPONENT source_s1

```

    PORT(P2, P1, P0 : IN  STD_LOGIC;
          D4, D3, D2,
          D1, D0    : IN  STD_LOGIC;
          lout       : OUT  STD_LOGIC;
          VDD, VSS   : INOUT STD_LOGIC);
END COMPONENT;
```

COMPONENT wpadin

```

    PORT(WORLD : IN  STD_LOGIC;
```

```

        OP : OUT STD_LOGIC);
END COMPONENT;

COMPONENT wpadout
  PORT(WORLD : OUT STD_LOGIC;
        IP : IN STD_LOGIC);
END COMPONENT;

COMPONENT wpadio
  PORT(WORLD : INOUT STD_LOGIC;
        IP : IN STD_LOGIC;
        EN : IN STD_LOGIC;
        OP : OUT STD_LOGIC);
END COMPONENT;

SIGNAL clk,
        por,
        shift_en, no_detect,
        shift_in,
        shift_out,
        test, test_detect,
        go, stim_detect,
        io_mode : STD_LOGIC;
-- *** pas de scan_chains
-- test_se,
-- test_si1,
-- test_si2,
SIGNAL source0, stim_source0 : STD_LOGIC_VECTOR(1 TO 4);
-- n1,n2,p1,p2: inactif = 3, sens1 = 9, sens2 = 6
SIGNAL amp_out, amp_in : STD_LOGIC_VECTOR(4 DOWNT0 0);
SIGNAL vref : STD_LOGIC;
SIGNAL output_s0_a,
        output_s0_b,
        output_s1_a : STD_LOGIC;

SIGNAL plage : STD_LOGIC_VECTOR(2 DOWNT0 0);

BEGIN
  module_stimulateur : stimulateur PORT MAP(
    clk => clk, por => por,
    shift_en => shift_en,
    shift_in => shift_in,
    shift_out => shift_out,
    test => test, go => go,
    no_detect => no_detect,
    stim_detect => stim_detect,
    test_detect => test_detect,
    stim_n1 => source0(1), stim_n2 => source0(2),
    stim_p1 => source0(3), stim_p2 => source0(4),
    amp => amp_out,
    plage => plage);

  module_source_s0 : source_s0 PORT MAP(
    stim_n1 => stim_source0(1), stim_n2 => stim_source0(2),
    stim_p1 => stim_source0(3), stim_p2 => stim_source0(4),
    D4 => amp_in(4),
    D3 => amp_in(3),
    D2 => amp_in(2),
    D1 => amp_in(1),
    D0 => amp_in(0),
    vref => vref,
    Iout_m => output_s0_a,
    Iout_p => output_s0_b);

```

```

module_source_s1 : source_s1 PORT MAP(
    P2 => plage(2),
    P1 => plage(1),
    P0 => plage(0),
    D4 => amp_in(4),
    D3 => amp_in(3),
    D2 => amp_in(2),
    D1 => amp_in(1),
    D0 => amp_in(0),
    Iout => output_s1_a);

-- PADS
pad_clk : wpadin PORT MAP( -- pad3
    WORLD => wclk,
    OP => clk);
pad_por : wpadin PORT MAP( -- pad4
    WORLD => wpor,
    OP => por);
pad_shift_en : wpadio PORT MAP( -- pad5
    WORLD => wshift_en,
    IP => no_detect,
    EN => io_mode,
    OP => shift_en);
pad_shift_in : wpadin PORT MAP( -- pad6
    WORLD => wshift_in,
    OP => shift_in);
pad_shift_out : wpadout PORT MAP( -- pad7
    WORLD => wshift_out,
    IP => shift_out);
pad_test : wpadio PORT MAP( -- pad8
    WORLD => wtest,
    IP => test_detect,
    EN => io_mode,
    OP => test);
pad_go : wpadio PORT MAP( -- pad9
    WORLD => wgo,
    IP => stim_detect,
    EN => io_mode,
    OP => go);
pad_io_mode : wpadin PORT MAP( -- pad10
    WORLD => wio_mode,
    OP => io_mode);
-- pa_test_se : wpadin PORT MAP( -- pad11
--     WORLD => wtest_se,
--     OP => test_se);
-- pad_test_si1 : wpadin PORT MAP( -- pad12
--     WORLD => wtest_si1,
--     OP => test_si1);
-- pad_test_si2 : wpadin PORT MAP( -- pad13
--     WORLD => wtest_si2,
--     OP => test_si2);
pad_n1 : wpadio PORT MAP( -- pad14
    WORLD => wn1,
    IP => source0(1),
    EN => io_mode,
    OP => stim_source0(1));
pad_n2 : wpadio PORT MAP( -- pad15
    WORLD => wn2,
    IP => source0(2),
    EN => io_mode,

```

```

        OP => stim_source0(2));
pad_p1 : wpadio PORT MAP( -- pad16
    WORLD => wp1,
    IP => source0(3),
    EN => io_mode,
    OP => stim_source0(3));
pad_p2 : wpadio PORT MAP( -- pad17
    WORLD => wp2,
    IP => source0(4),
    EN => io_mode,
    OP => stim_source0(4));
pad_amp4 : wpadio PORT MAP( -- pad18
    WORLD => wamp(4),
    IP => amp_out(4),
    EN => io_mode,
    OP => amp_in(4));
pad_amp3 : wpadio PORT MAP( -- pad19
    WORLD => wamp(3),
    IP => amp_out(3),
    EN => io_mode,
    OP => amp_in(3));
pad_amp2 : wpadio PORT MAP( -- pad20
    WORLD => wamp(2),
    IP => amp_out(2),
    EN => io_mode,
    OP => amp_in(2));
pad_amp1 : wpadio PORT MAP( -- pad21
    WORLD => wamp(1),
    IP => amp_out(1),
    EN => io_mode,
    OP => amp_in(1));
pad_amp0 : wpadio PORT MAP( -- pad22
    WORLD => wamp(0),
    IP => amp_out(0),
    EN => io_mode,
    OP => amp_in(0));
pad_vref : wpadin PORT MAP( -- pad23
    WORLD => wvref,
    OP => vref);
pad_out0a : wpadout PORT MAP( -- pad24
    WORLD => woutput_s0_a,
    IP => output_s0_a);
pad_out0b : wpadout PORT MAP( -- pad25
    WORLD => woutput_s0_b,
    IP => output_s0_b);
pad_out1a : wpadout PORT MAP( -- pad26
    WORLD => woutput_s1_a,
    IP => output_s1_a);

END struct;

```

Annexe II

Schéma électrique et code assembleur du stimulateur externe portable

II.1 Schéma électrique

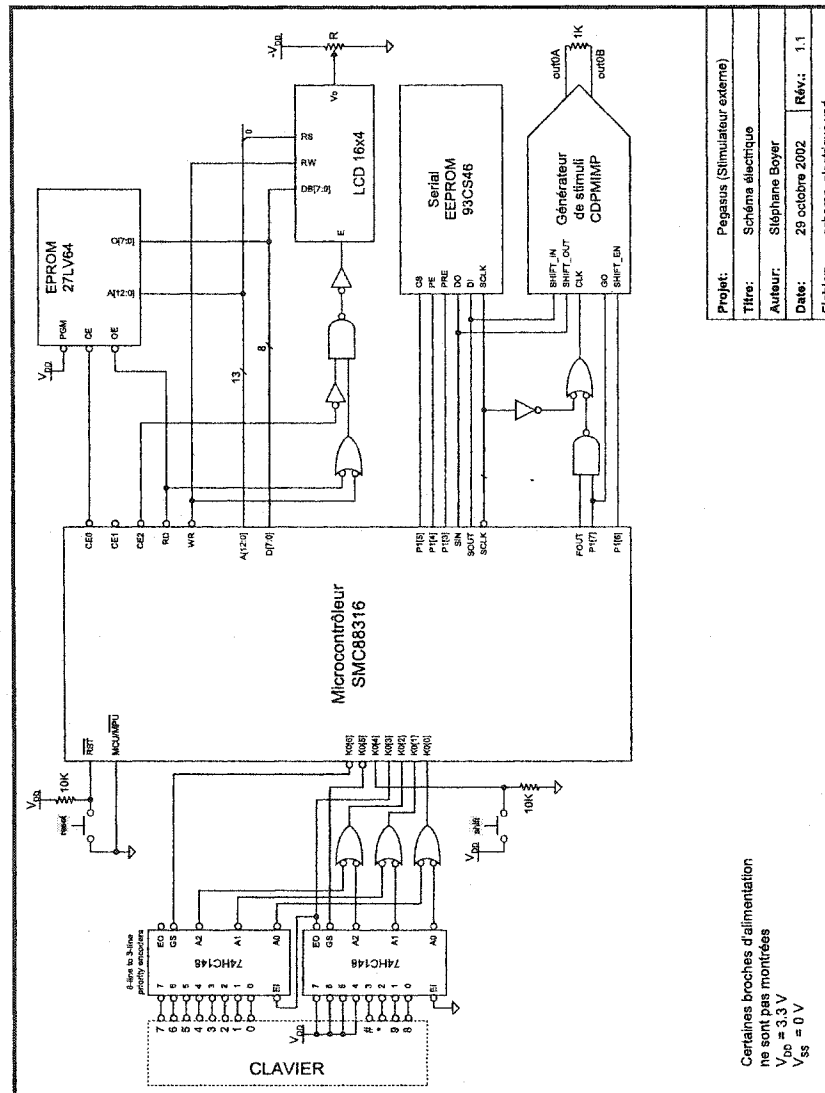


Figure II.1: Schéma électrique du stimulateur neuromusculaire externe et portable.

II.2 Code assembleur du stimulateur externe

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:          Pegasus (Stimulateur externe)
-----
; Fichier:         "PEGASUS.ASM"
;
; Description:
;   Programme principal du stimulateur externe
;
;
; Date:            3 juillet 2002
;
; Auteur:          Stephane Boyer
-----
; Processeur:      SMC88316 (S-MOS)
-----
CPU "SMC88316.TBL"      ; TABLE DES INSTRUCTIONS
HOF "MOT8"              ; FORMAT HEXADECIMAL INTEL
; PAGE 55              ; 55 lignes par page
TITL "Pegasus (Stimulateur externe)"
;
-----
; *** Constantes ***
RAM_Debut: EQU 0F000h    ; Debut de la RAM interne
RAM_Fin:   EQU 0F800h    ; Adresse initiale du Stack Pointer
                          ; Pointeur de pile pre-decremente
;
; *** Variables ***
ORG RAM_Debut
fenetre_active: DFS 1    ; Menu affiche
fenetre_old:   DFS 1    ; Memoire pour la derniere fenetre
clavier_entree: DFS 1    ; Valeur entree au clavier
; lp:          DFS 1    ; Longueur du parametre
tmp3:          DFS 1    ; Nouvelle valeur du parametre en BCD
tmp2:          DFS 1    ; Nouvelle valeur du parametre en BCD
tmp1:          DFS 1    ; Nouvelle valeur du parametre en BCD
tmp0:          DFS 1    ; Nouvelle valeur du parametre en BCD
tmpvalid:      DFS 1    ; Nouvelle valeur du parametre valide (0, 1, FF)
;
; Structure de l'ensemble des parametres de stimulation
;
oefrq3:        DFS 1
oefrq2:        DFS 1
oefrq1:        DFS 1
oefrq0:        DFS 1    ; Frequence de l'onde excitante
oeamp1:        DFS 1
oeamp0:        DFS 1    ; Amplitude de l'onde excitante
oepwd2:        DFS 1
oepwd1:        DFS 1
oepwd0:        DFS 1    ; Largeur d'impulsion de l'onde excitante
oeipd2:        DFS 1
oeipd1:        DFS 1
oeipd0:        DFS 1    ; Delai inter-impulsion de l'onde excitante
oifrq3:        DFS 1
oifrq2:        DFS 1
oifrq1:        DFS 1
oifrq0:        DFS 1    ; Frequence de l'onde inhibitrice

```

```

oiamp1:      DFS 1
oiamp0:      DFS 1      ; Amplitude de l'onde inhibitrice
oipwd2:      DFS 1
oipwd1:      DFS 1
oipwd0:      DFS 1      ; Largeur d'impulsion de l'onde inhibitrice
oiipd2:      DFS 1
oiipd1:      DFS 1
oiipd0:      DFS 1      ; Delai inter-impulsion de l'onde inhibitrice
duree1:      DFS 1
duree0:      DFS 1      ; Duree de la stimulation

; Variable pour les operations du sEEPROM
memadr:      DFS 1      ; Adresse du sEEPROM (6 bits)
memMSB:      DFS 1      ; MSB du sEEPROM (donnees de 16 bits)
memLSB:      DFS 1      ; LSB du sEEPROM (donnees de 16 bits)

; Variable temporaire pour l'affichage des donnees en memoire
memoire:     DFS 1      ; Ensemble de parametres a charger
oeqr3t:      DFS 1
oeqr2t:      DFS 1
oeqr1t:      DFS 1
oeqr0t:      DFS 1      ; Frequence de l'onde excitante
oeamp1:      DFS 1
oeamp0t:     DFS 1      ; Amplitude de l'onde excitante
oepwd2t:     DFS 1
oepwd1t:     DFS 1
oepwd0t:     DFS 1      ; Largeur d'impulsion de l'onde excitante
oeipd2t:     DFS 1
oeipd1t:     DFS 1
oeipd0t:     DFS 1      ; Delai inter-impulsion de l'onde excitante
oifrq3t:     DFS 1
oifrq2t:     DFS 1
oifrq1t:     DFS 1
oifrq0t:     DFS 1      ; Frequence de l'onde inhibitrice
oiamp1t:     DFS 1
oiamp0t:     DFS 1      ; Amplitude de l'onde inhibitrice
oipwd2t:     DFS 1
oipwd1t:     DFS 1
oipwd0t:     DFS 1      ; Largeur d'impulsion de l'onde inhibitrice
oiipd2t:     DFS 1
oiipd1t:     DFS 1
oiipd0t:     DFS 1      ; Delai inter-impulsion de l'onde inhibitrice

;-----
; Initialisation generale
;
ORG 0
INCL "W:\88316REG.ASM"      ; Vecteurs d'exception
INCL "W:\88316RST.ASM"     ; Initialisation des registres
;
;-----
; Initialisation des variables
LDA  #OFFh
STA  clavier_entree
BSR  Mem_Init              ; Initialisation des parametres de stimulation

; Initialisation des peripheriques (LCD, clavier)
BSR  LCD_Init              ; Initialisation du LCD
BSR  Int_K04_K07_Init      ; Init. des interruptions pour le clavier

; Message d'initialisation
LDA  #00h
STA  fenetre_active

```

```

LDX  #Aff_Init
LDY  #Var_Init
BSR  LCD_Update

; Pause de deux secondes
LDA  #010h
STA  Timer_Mode          ; 16-bit mode, fosc1
LDA  #00h
STA  Timer0_Latch
LDA  #004h
STA  Timer1_Latch        ; 4 * 256
LDA  #080h
STA  External_IFR        ; Reset Interrupt Flag Timer1
LDA  #01Bh
STA  Timer0_Mode          ; fosc1 / 16, one-shot, run
Init_delai:
LDA  External_IFR
CMPA #080h
BLT  Init_delai          ; Attendre le facteur d'interruption
;
; -----
; *** Programme principal ***
;
Principal:
LDA  #01h
STA  fenetre_active
LDX  #Aff_Main            ; Afficher la fenetre principale
BSR  LCD_Update

Lire_clavier:
LDA  clavier_entree
CMPA #OFFh
BEQ  Lire_clavier        ; Aucune entree au clavier
; LDA  #00h
; STA  PortP1_Data        ; Stimulation inactive
; BEQ  Stim_inactive
;
LDZ  #External_IER        ; Desactiver les interruptions du clavier
ANDM #OEFh
;
BSR  Commande            ; Traitement de l'entree au clavier
BSR  LCD_Update          ; Mise a jour du LCD
LDA  #OFFh
STA  clavier_entree      ; Reinitialiser la variable clavier_entree
;
LDZ  #External_IER        ; Reactiver les interruptions du clavier
ORM  #010h
BRA  Lire_clavier

; -----
; Gestion de l'interruption Watch Dog
; Oscillation de R50
Int_Watch_Dog:
PHALE                                ; Sauvegarde de tous les registres
LDA  PortR5_Data
XORA #001h                          ; Modifier R50 seulement
STA  PortR5_Data                    ; Oscillation du port R50
PLALE                                ; Recuperation des registres
RTI
;
; -----
; *** Sous-routines et tableaux ***
;

```

```

INCL "W:\CLAVIER.ASM"      ; Interpretation du clavier
INCL "W:\COMMANDE.ASM"     ; Interpretation des entrees usager
INCL "W:\MEMOIRE.ASM"      ; Routines pour la gestion des parametres
INCL "W:\LCD.ASM"          ; Routines pour l'affichage sur le LCD
INCL "W:\LCD_TBL.ASM"      ; Tableau du texte a afficher sur le LCD

```

```

;

```

```

; Definition des registres pour les ports d'entree et de sortie

```

```

INCL "W:\88316_IO.ASM"

```

```

;

```

```

; Adresse du LCD (A0 = RS)

```

```

ORG 004000h                ; Voir "pegasus/doc/memory_map.txt"

```

```

LCD_Command:      DFS    1      ; RS=0

```

```

LCD_Data:         DFS    1      ; RS=1

```

```

;

```

```

END

```

```

; Fin de "PEGASUS.ASM"

```

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:          Pegasus (Stimulateur externe)
-----
; Fichier:         "88316REG.ASM"
;
; Description:
;   Definitions des registres et des vecteurs
;   d'interruption du SMC88316 (S-MOS)
;
;
; Date:           26 mars 2002
;
; Auteur:         G.-E. April
; Modifications:   Stephane Boyer
-----
; Processeur:      SMC88316 (S-MOS)
-----
;
ORG 0
;
Reset_Vector:      DWL Reset          ; 88316RST.ASM
Div0_Vector:       DWL Dummy
Watch_Dog_Vector:  DWL Int_Watch_Dog ; PEGASUS.ASM
T1_Int_Vector:     DWL Dummy
T2_Int_Vector:     DWL Dummy
K10_11_Int_Vector: DWL Dummy
K04_07_Int_Vector: DWL Int_K04_K07   ; CLAVIER.ASM
K00_03_Int_Vector: DWL Dummy
SIO_Err_Int_Vector: DWL Dummy
SIO_Rcv_Int_Vector: DWL Dummy
SIO_Snd_Int_Vector: DWL Dummy
Stop_Watch_100Hz_Vector: DWL Dummy
Stop_Watch_010Hz_Vector: DWL Dummy
Stop_Watch_001Hz_Vector: DWL Dummy
Clock_32Hz_Vector:  DWL Dummy
Clock_08Hz_Vector:  DWL Dummy
Clock_02Hz_Vector:  DWL Dummy
Clock_01Hz_Vector:  DWL Int_01Hz     ; MEMOIRE.ASM
    Org **2          ; Reserved location
SWI_019_Vector:     DWL Dummy
;
SWI_020_Vector:     DWL Dummy
SWI_021_Vector:     DWL Dummy
SWI_022_Vector:     DWL Dummy
SWI_023_Vector:     DWL Dummy
SWI_024_Vector:     DWL Dummy
SWI_025_Vector:     DWL Dummy
SWI_026_Vector:     DWL Dummy
SWI_027_Vector:     DWL Dummy
SWI_028_Vector:     DWL Dummy
SWI_029_Vector:     DWL Dummy
;
SWI_030_Vector:     DWL Dummy
SWI_031_Vector:     DWL Dummy
SWI_032_Vector:     DWL Dummy
SWI_033_Vector:     DWL Dummy
SWI_034_Vector:     DWL Dummy
SWI_035_Vector:     DWL Dummy
SWI_036_Vector:     DWL Dummy

```

SWI_037_Vector:	DWL	Dummy
SWI_038_Vector:	DWL	Dummy
SWI_039_Vector:	DWL	Dummy
;		
SWI_040_Vector:	DWL	Dummy
SWI_041_Vector:	DWL	Dummy
SWI_042_Vector:	DWL	Dummy
SWI_043_Vector:	DWL	Dummy
SWI_044_Vector:	DWL	Dummy
SWI_045_Vector:	DWL	Dummy
SWI_046_Vector:	DWL	Dummy
SWI_047_Vector:	DWL	Dummy
SWI_048_Vector:	DWL	Dummy
SWI_049_Vector:	DWL	Dummy
;		
SWI_050_Vector:	DWL	Dummy
SWI_051_Vector:	DWL	Dummy
SWI_052_Vector:	DWL	Dummy
SWI_053_Vector:	DWL	Dummy
SWI_054_Vector:	DWL	Dummy
SWI_055_Vector:	DWL	Dummy
SWI_056_Vector:	DWL	Dummy
SWI_057_Vector:	DWL	Dummy
SWI_058_Vector:	DWL	Dummy
SWI_059_Vector:	DWL	Dummy
;		
SWI_060_Vector:	DWL	Dummy
SWI_061_Vector:	DWL	Dummy
SWI_062_Vector:	DWL	Dummy
SWI_063_Vector:	DWL	Dummy
SWI_064_Vector:	DWL	Dummy
SWI_065_Vector:	DWL	Dummy
SWI_066_Vector:	DWL	Dummy
SWI_067_Vector:	DWL	Dummy
SWI_068_Vector:	DWL	Dummy
SWI_069_Vector:	DWL	Dummy
;		
SWI_070_Vector:	DWL	Dummy
SWI_071_Vector:	DWL	Dummy
SWI_072_Vector:	DWL	Dummy
SWI_073_Vector:	DWL	Dummy
SWI_074_Vector:	DWL	Dummy
SWI_075_Vector:	DWL	Dummy
SWI_076_Vector:	DWL	Dummy
SWI_077_Vector:	DWL	Dummy
SWI_078_Vector:	DWL	Dummy
SWI_079_Vector:	DWL	Dummy
;		
SWI_080_Vector:	DWL	Dummy
SWI_081_Vector:	DWL	Dummy
SWI_082_Vector:	DWL	Dummy
SWI_083_Vector:	DWL	Dummy
SWI_084_Vector:	DWL	Dummy
SWI_085_Vector:	DWL	Dummy
SWI_086_Vector:	DWL	Dummy
SWI_087_Vector:	DWL	Dummy
SWI_088_Vector:	DWL	Dummy
SWI_089_Vector:	DWL	Dummy
;		
SWI_090_Vector:	DWL	Dummy
SWI_091_Vector:	DWL	Dummy
SWI_092_Vector:	DWL	Dummy
SWI_093_Vector:	DWL	Dummy

```

SWI_094_Vector:      DWL  Dummy
SWI_095_Vector:      DWL  Dummy
SWI_096_Vector:      DWL  Dummy
SWI_097_Vector:      DWL  Dummy
SWI_098_Vector:      DWL  Dummy
SWI_099_Vector:      DWL  Dummy
;
SWI_100_Vector:      DWL  Dummy
SWI_101_Vector:      DWL  Dummy
SWI_102_Vector:      DWL  Dummy
SWI_103_Vector:      DWL  Dummy
SWI_104_Vector:      DWL  Dummy
SWI_105_Vector:      DWL  Dummy
SWI_106_Vector:      DWL  Dummy
SWI_107_Vector:      DWL  Dummy
SWI_108_Vector:      DWL  Dummy
SWI_109_Vector:      DWL  Dummy
;
SWI_110_Vector:      DWL  Dummy
SWI_111_Vector:      DWL  Dummy
SWI_112_Vector:      DWL  Dummy
SWI_113_Vector:      DWL  Dummy
SWI_114_Vector:      DWL  Dummy
SWI_115_Vector:      DWL  Dummy
SWI_116_Vector:      DWL  Dummy
SWI_117_Vector:      DWL  Dummy
SWI_118_Vector:      DWL  Dummy
SWI_119_Vector:      DWL  Dummy
;
SWI_120_Vector:      DWL  Dummy
SWI_121_Vector:      DWL  Dummy
SWI_122_Vector:      DWL  Dummy
SWI_123_Vector:      DWL  Dummy
SWI_124_Vector:      DWL  Dummy
SWI_125_Vector:      DWL  Dummy
SWI_126_Vector:      DWL  Dummy
SWI_127_Vector:      DWL  Dummy
;
;
Dummy:  RTI          ; Aucune operation
;
;   Fin de "88316REG.ASM"
;-----

```

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:          Pegasus (Stimulateur externe)
-----
; Fichier:          "88316RST.ASM"
;
; Description:
;   Sequence d'initialisation du SMC88316.
;   Osc1/8 sur Fout
;
; Date:            26 mars 2002
;
; Auteur:          G.-E. April
; Modifications:    Stephane Boyer
-----
; Processeur: SMC88316 (S-MOS)
-----
;
Reset:
    LDA #00h          ; Data register
    LDB #00h          ; Data register
    LDL #00h          ; Index data register
    LDH #00h          ; Index data register
    LDX #00h          ; Index register IX
    LDY #00h          ; Index register IY
    LDSP #RAM_Fin     ; Stack Pointer
    LDDPR #00h        ; Base register
    LDEP #00h         ; Extended page register
    LDXP #00h         ; Extended Index X register
    LDYP #00h         ; Extended Index Y register
;
    LDA #045h
    STA Bus_mode      ; Bus (64K), Chip (8K), CEO et CE2 enabled
    LDA #00h
    STA Stack_Pointer_Page ; SP init. pour activer les interruptions
    LDA #3Fh
    STA Fout_Ctrl     ; Fout ON, OSC1/8, Clock timer ON
;   lda #00h
;   sta LCD_Mode      ; 1/32, 5*8 matrix
;   lda #0
;   sta LCD_Control   ; Driver OFF
;   lda #0
;   sta SVD           ; Disabled
;   lda #0
;   sta Comp_Ctrl     ; Comparator disabled.
;   lda #041h
;   sta Priority_Ctrl0 ; K07_K00 (L1), Clock (L1)
;   lda #0
;   sta Priority_Ctrl1 ; All disabled for now
;   lda #0
;   sta Clock_IER     ; All disabled for now
;   lda #010h
;   sta External_IER  ; K07_K00 interrupt enabled
;   lda #OFFH
;   sta Clock_IFR     ; Clear flags
;   lda #OFFH
;   sta External_IFR  ; Clear flags
;   lda #0
;   sta Timer_Mode    ; Tout disabled
;   lda #0

```

```

; sta Timer0_Mode      ; Disabled
; lda #0
; sta Timer1_Mode      ; Disabled
; lda #100
; sta Timer0_Latch     ; Count 100
; lda #100
; sta Timer1_Latch     ; Count 100
; lda #03FH
; sta Fout_Ctrl        ; Fout is OSC1/8, Clock Run
; lda #0
; sta StopWatch_Ctrl   ; StopWatch stopped
; lda #0
; sta Buzzer_Ctrl0     ; Buzzer OFF
; lda #0
; sta Buzzer_Ctrl1     ; Default mode
; lda #0
; sta SIO_Ctrl         ; Disabled
; lda #0
; sta SIO_Status       ; Disabled
; lda #080h
; sta KO_IER           ; K07 Interrupt enable
; lda #0
; sta K1_IER           ; K1 Interrupts disabled
; lda #0
; sta KO_IPR           ; K0 Interrupts on rising edge
; lda #0FFh
; sta K1_IPR           ; K1 Interrupts on falling edge
; lda #0FFh
; sta PortP1_DDR       ; Output
; lda #0
; sta PortR0_1_5_Ctrl  ; Complementary
;
; Fin de "88316RST.ASM"
-----

```

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:      Stimulateur externe
-----
; Fichier:      "88316_io.asm"
;
; Description:
;   Definitions des ports d'entree/sortie du SMC88316 (S-MOS)
;
;
; Date:         11 octobre 1999
;
; Auteur:       G.-E. April
; Modifications: S. Boyer
-----
; Processeur: SMC88316 (S-MOS)
-----
;
;   ORG 00FF00H
;
Bus_Mode:      DFS      1
;   Bits 7 et 6   Bus mode
;   00           Single chip
;   01           64K expansion
;   10           512K (Minimum) expansion
;   11           512K (Maximum) expansion
;   Bits 5 et 4   CE mode
;   00           8K
;   01           16K
;   10           32K
;   11           64K
;   Bit3         CE3 enable
;   Bit2         CE2 enable
;   Bit1         CE1 enable
;   Bit0         CE0 enable
;
;
Stack_Pointer_Page:  DFS      1
;   Only in expanded mode
;
Speed_Control:  DFS      1          ; (OFF02H)
;   Bit 7       Bus release enable
;   Bits 6 to 4   Wait control
;   000         no wait
;   001         2 states
;   010         4 states
;   011         6 states
;   100         8 states
;   101         10 states
;   110         12 states
;   111         14 states
;   Bit 3       Osc3 used as system clock (else, Osc1)
;   Bit 2       Osc3 Running
;   Bits 1 et 0   Operating mode
;   00          Normal (Vdd>2.2 volts)
;   01          Low power (Vdd>1.3 volts)
;   1x          High speed (Vdd>3.3 volts)
;
;
;   ORG 00FF10H

```

```

;
LCD_Mode:      DFS      1
;      Bit 7 to 5 not implemented (=0)
;      Bit 4      External driver CLock output enable
;      Bit 3      External driver FFrame output enable
;      Bit 2      5x5 Font format (Else, 5x8)
;      Bit 1      1/16 Duty (Else, 1/32)
;      Bit 0      Reserved
;
LCD_Control:   DFS      1
;      Bit 7      not implemented (=0)
;      Bit 6      Display memory area 1 (else 0)
;      Bits 5 & 4  Display control
;                  00      Driver off
;                  01      Normal display
;                  10      All dots off (blank)
;                  11      All dots on
;      Bits 3 to 0 Contrast (High=Dark)
;
SVD:           DFS      1      ; Supply voltage detection.
;      Bit 7 & 6 not implemented (=0)
;      Bit 5      Auto sampling enable
;      Bit 4      Write: Continuous sampling enable
;                  Read:  Busy (else ready)
;      Bits 3 to 0 Detection level (0 to 14)
;
Comp_Ctrl:     DFS      1      ; (OFF13H)
;      Bits 7 to 4 not implemented (=0)
;      Bit 3      Comparator 1 enable
;      Bit 2      Comparator 0 enable
;      Bit 1      Comparator 1 data
;      Bit 0      Comparator 0 data
;
;
;      ORG 00FF20H
;
Priority_Ctrl0: DFS      1
;      Bits 7 & 6  K00 to K07
;      Bits 5 & 4  SIO
;      Bits 3 & 2  Stopwatch
;      Bits 1 & 0  Clock   (Pins 95 to 88)
;
Priority_Ctrl1: DFS      1
;      Bits 7 to 4 not implemented (=0)
;      Bits 3 & 2  Timer
;      Bits 1 & 0  K10 and K11
;
Clock_IER:     DFS      1      ; (OFF22H)
;      Bit 7      not implemented (=0)
;      Bit 6      Stopwatch 100 Hz enable
;      Bit 5      Stopwatch 10 Hz enable
;      Bit 4      Stopwatch 1 Hz enable
;      Bit 3      Clock 32 Hz enable
;      Bit 2      Clock 8 Hz enable
;      Bit 1      Clock 2 Hz enable
;      Bit 0      Clock 1 Hz enable
;
External_IER:  DFS      1      ; (OFF23H)
;      Bit 7      Timer1 interrupt enable
;      Bit 6      Timer0 interrupt enable
;      Bit 5      K10 & K11 interrupt enable (Pins 87 & 86)
;      Bit 4      K04 to K07 interrupt enable (Pins 91 to 88)
;      Bit 3      K00 to K03 interrupt enable (Pins 95 to 92)

```

```

;      Bit 2   SIO error interrupt enable
;      Bit 1   SIO receive interrupt enable
;      Bit 0   SIO send interrupt enable
;
Clock_IFR:    DFS      1
;      Bit 7 not implemented (=0)
;      Bit 6   Stopwatch 100 Hz flag
;      Bit 5   Stopwatch 10 Hz flag
;      Bit 4   Stopwatch 1 Hz flag
;      Bit 3   Clock 32 Hz flag
;      Bit 2   Clock 8 Hz flag
;      Bit 1   Clock 2 Hz flag
;      Bit 0   Clock 1 Hz flag
;
External_IFR: DFS      1      ; (OFF25H)
;      Bit 7   Timer1 interrupt flag
;      Bit 6   Timer0 interrupt flag
;      Bit 5   K10 & K11 interrupt flag (Pins 86 & 87)
;      Bit 4   K04 to K07 interrupt flag (Pins 91 to 98)
;      Bit 3   K00 to K03 interrupt flag (Pins 95 to 92)
;      Bit 2   SIO error interrupt flag
;      Bit 1   SIO receive interrupt flag
;      Bit 0   SIO send interrupt flag
;
;
      ORG 00FF30H
;
Timer_Mode:   DFS      1
;      Bits 7 to 5 not implemented (=0)
;      Bit 4   Single 16 bit timer (else, 2x 8 bits)
;      Bit 3   TOUT= Timer 1 (else, Timer 0)
;      Bit 2   Enable TOUT
;      Bit 1   Prescaler 1 clock = Osci3 (else, Osci1)
;      Bit 0   Prescaler 0 clock = Osci3 (else, Osci1)
;
Timer0_Mode:  DFS      1
;      Bit 7   Enable event counter mode
;      Bit 6   Enable pulse width measurement and noise reject.
;      Bit 5   Pulse polarity or level selection
;      Bits 4 & 3 Prescaler divide ratio
;               00      /1
;               01      /4
;               10      /16
;               11      /64
;      Bit 2   Continuous/One-shot)
;      Bit 1   Preset (reads as 0)
;      Bit 0   Run/Stop
;
Timer1_Mode:  DFS      1
;      Bits 7 to 5 not implemented (=0)
;      Bits 4 & 3 Prescaler divide ratio
;               00      /1
;               01      /4
;               10      /16
;               11      /64
;      Bit 2   Continuous/One-shot)
;      Bit 1   Preset (reads as 0)
;      Bit 0   Run/Stop
;
Timer0_Latch: DFS      1      ; (OFF33h)
;
Timer1_Latch: DFS      1      ; (OFF34h)
;

```

```

Timer0_Counter: DFS      1      ; (OFF35h)
;      (Read only)
;
Timer1_Counter: DFS      1      ; (OFF36h)
;      (Read only)
;
;
;      ORG 00FF40H
;
FOUT_Ctrl:      DFS      1      ; Fout = Pin 140 (R34)
;      Bit 7 not implemented (=0)
;      Bits 6 to 4      FOUT frequency selection
;          000      Osc1/1
;          001      Osc1/2
;          010      Osc1/4
;          011      Osc1/8
;          100      Osc3/1
;          101      Osc3/2
;          110      Osc3/4
;          111      Osc3/8
;      Bit 3      Enable FOUT
;      Bit 2      Watchdog Reset (reads as 0)
;      Bit 1      Clock Reset (reads as 0)
;      Bit 0      Clock Run/Stop_
;
Clock_Data:      DFS      1      ; (OFF41H)
;      Bit 7      1 Hz      (read only)
;      Bit 6      2 Hz      (read only)
;      Bit 5      4 Hz      (read only)
;      Bit 4      8 Hz      (read only)
;      Bit 3      16 Hz     (read only)
;      Bit 2      32 Hz     (read only)
;      Bit 1      64 Hz     (read only)
;      Bit 0      128 Hz    (read only)
;
StopWatch_Ctrl: DFS      1
;      Bits 2 to 7 not implemented (=0)
;      Bit 1      Stopwatch Reset
;      Bit 0      Stopwatch Run/Stop
;
StopWatch_Data: DFS      1
;      Bits 7 to 4      Tenhs of a second
;      Bits 3 to 0      Hundreths of a second
;
Buzzer_Ctrl0:   DFS      1
;      Bit 7      not implemented (=0)
;      Bit 6      one-shot buzzer stop
;      Bit 5      one-shot buzzer trigger/status
;      Bit 4      one-shot buzzer duration
;      Bit 3      Envelope attenuation time
;      Bit 2      Envelope reset
;      Bit 1      Envelope on
;      Bit 0      Buzzer on
;
Buzzer_Ctrl1:   DFS      1
;      Bit 7      not implemented (=0)
;      Bits 6 to 4      Buzzer duty ratio
;      Bit 3          Not implemented (=0)
;      Bits 2 to 0      Frequency
;
;
;      ORG 00FF48H
;

```

```

SIO_Ctrl:      DFS      1
;      Bit 7 not implemented (=0)
;      Bit 6 Parity enable
;      Bit 5 Odd parity (else, even)
;      Bits 4 & 3 Clock source selection
;      00 Osc3/16
;      01 Osc3/8
;      10 Osc3/4
;      11 Timer
;      Bits 2 & 1 SIO mode
;      00 Synchronounous master
;      01 Synchronounous Slave
;      10 7 bit asynchronous
;      11 8 bit asynchronous
;      Bit 0 SIO enable (else, regular I/O bit)
;
SIO_Status:    DFS      1
;      Bit 7 not implemented (=0)
;      Bit 6 Framing error
;      Bit 5 Parity error
;      Bit 4 Overrun error
;      Bit 3 Receive status (read) or trigger (write)
;      Bit 2 Receive enable
;      Bit 1 Transmit status (read) or trigger (write)
;      Bit 0 Transmit enable
;
SIO_Data:      DFS      1
;
;
;      ORG 00FF50H
;
KO_IER:        DFS      1
;      Individual enables for each K0x interrupt
;      Bit 7 K07 IE (Pin 88)
;      Bit 6 K06 IE (Pin 89)
;      Bit 5 K05 IE (Pin 90)
;      Bit 4 K04 IE (Pin 91)
;      Bit 3 K03 IE (Pin 92)
;      Bit 2 K02 IE (Pin 93)
;      Bit 1 K01 IE (Pin 94)
;      Bit 0 K00 IE (Pin 95)
;
K1_IER:        DFS      1
;      Bits 7 to 2 not implemented (=0)
;      Bit 1 Enable K11 (Pin 86)
;      Bit 0 Enable K10 (Pin 87)
;
KO_IPR:        DFS      1 ; Polarity selection (1=Falling)
;      Individual selection for each K0x interrupt
;      Bit 7 K07 IPol (Pin 88)
;      Bit 6 K06 IPol (Pin 89)
;      Bit 5 K05 IPol (Pin 90)
;      Bit 4 K04 IPol (Pin 91)
;      Bit 3 K03 IPol (Pin 92)
;      Bit 2 K02 IPol (Pin 93)
;      Bit 1 K01 IPol (Pin 94)
;      Bit 0 K00 IPol (Pin 95)
;
K1_IPR:        DFS      1 ; Polarity selection (1=Falling)
;      Bits 7 to 2 not implemented (=0)
;      Bit 1 Select K11 falling edge (Pin 86)
;      Bit 0 Select K10 falling edge (Pin 87)
;

```

```

KO_Data:      DFS      1      ; 8 bits of input data
;      Bit 7      K07 Datum      (Pin 88)
;      Bit 6      K06 Datum      (Pin 89)
;      Bit 5      K05 Datum      (Pin 90)
;      Bit 4      K04 Datum      (Pin 91)
;      Bit 3      K03 Datum      (Pin 92)
;      Bit 2      K02 Datum      (Pin 93)
;      Bit 1      K01 Datum      (Pin 94)
;      Bit 0      K00 Datum      (Pin 95)
;
K1_Data:      DFS      1      ; (OFF55H)
;      Bits 7 to 2 not implemented (=0)
;      Bit 1      Data K11/BusRequest* (Pin 86)
;      Bit 0      Data K10/Event_Input (Pin 87)
;
;
      ORG 00FF60H
;
PortP0_DDR:    DFS      1      ; (1 = output)
;
PortP1_DDR:    DFS      1      ; (1 = output)
;      Bit 7      P17 Dir (Pin 96)
;      Bit 6      P16 Dir (Pin 97)
;      Bit 5      P15 Dir (Pin 98)
;      Bit 4      P14 Dir (Pin 99)
;      Bit 3      P13 Dir (Pin 100)
;      Bit 2      P12 Dir (Pin 101)
;      Bit 1      P11 Dir (Pin 102)
;      Bit 0      P10 Dir (Pin 103)
;
PortP0_Data:   DFS      1      ; (OFF62H)
;
PortP1_Data:   DFS      1      ; (OFF63H)
;      Bit 7      P17 Datum (Comparator1 -) (Pin 96)
;      Bit 6      P16 Datum (Comparator1 +) (Pin 97)
;      Bit 5      P15 Datum (Comparator0 -) (Pin 98)
;      Bit 4      P14 Datum (Comparator0 +) (Pin 99)
;      Bit 3      P13 Datum (Serial Ready*) (Pin 100)
;      Bit 2      P12 Datum (Serial Clock*) (Pin 101)
;      Bit 1      P11 Datum (Serial Output) (Pin 102)
;      Bit 0      P10 Datum (Serial Input) (Pin 103)
;
;
      ORG 00FF70H
;
PortR0_1_5_Ctrl: DFS      1      ; (1 = High-Z)
;      Bit 7      Port R51
;      Bit 6      Port R50
;      Bit 5      R/W register
;      Bit 4      R/W register
;      Bit 3      Ports R17 to R14 (R/W register)
;      Bit 2      Ports R13 to R10 (R/W register)
;      Bit 1      Ports R07 to R04 (R/W register)
;      Bit 0      Ports R03 to R00 (R/W register)
;
PortR2_Ctrl:   DFS      1      ; (1 = High-Z)
;      Individual controls for Ports R27 to R20
;
PortR3_Ctrl:   DFS      1      ; (1 = High-Z)
;      Individual controls for Ports R37 to R30
;      Bit 7      Port R37
;      Bit 6      Port R36
;      Bit 5      Port R35

```

```

;      Bit 4   Port R34
;      Bit 3   Port R33
;      Bit 2   Port R32
;      Bit 1   Port R31
;      Bit 0   Port R30
;
;
PortR0_Data:   DFS      1
;      Datum for Port R07 to R00      (Pins 119 to 112)
;
PortR1_Data:   DFS      1
;      Datum for Port R17 to R10      (Pins 127 to 120)
;
PortR2_Data:   DFS      1
;      Bit 7   R27 Datum (Tout) (Pin 135)
;      Bit 6   R26 Datum (FR)   (Pin 134)
;      Bit 5   R25 Datum (CL)   (Pin 133)
;      Bit 4   R24 Datum (WR\ ) (Pin 132)
;      Bit 3   R23 Datum (RD\ ) (Pin 131)
;      Bit 2   R22 Datum        (Pin 130)
;      Bit 1   R21 Datum        (Pin 129)
;      Bit 0   R20 Datum        (Pin 128)
;
PortR3_Data:   DFS      1
;      Bit 4   R34 Datum (Fout)   (Pin 140)
;      Bit 3   R33 Datum (CE3\ ) (Pin 139)
;      Bit 2   R32 Datum (CE2\ ) (Pin 138)
;      Bit 1   R31 Datum (CE1\ ) (Pin 137)
;      Bit 0   R30 Datum (CE0\ ) (Pin 136)
;
PortR4_Data:   DFS      1      ; Internal register only
;
PortR5_Data:   DFS      1      ; (OFF78H)
;      Bits 7 to 2 not implemented (=0)
;      Bit 1   R51/Bus_Acknowledge\ (Pin 142)
;      Bit 0   R50/Buzzer           (Pin 141)
;
;
;      Fin de "88316_io.asm"
;-----

```

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:          Pegasus (Stimulateur externe)
-----
; Fichier:         "CLAVIER.ASM"
;
; Description:
;   Gere le port K0 sur lequel est branche le clavier.
;   Comprend les sous-routines:
;       Int_K04_K07_Init
;           . Initialisation des exceptions pour le clavier
;
;       Int_K04_K07
;           . Exception pour la gestion des entrees au clavier.
;           . Retour de la valeur dans la variable clavier_entree.
;           . Retourne la valeur binaire pour les chiffre [0-9].
;           . Retourne #0Ah pour l'etoile.
;           . Retourne #0Bh pour le carre (#).
;           . La touche shift est sur le bit 4 (#00010000b) et ne
;             genere pas d'interruption.
;
; Auteur:          Stephane Boyer
;
; Date:            3 juillet 2002
-----
;
; *** Variables utilisees ***
; clavier_entree          ; Entree du clavier
;
;
; Activation des interruption sur les ports K05-K06 et Clock_01Hz
Int_K04_K07_Init:
    LDA    #060h
    STA    K0_IER          ; Activation des interruptions sur K06-K05
    LDA    #0FFh
    STA    K0_IPR          ; Interruptions sur le front descendant
    LDA    #080h
    STA    Priority_Ctrl0   ; Interruptions K07-K00, IRQ2
    LDA    #0FFh
    STA    External_IFR     ; Reinitialisation des facteurs d'interruption
    LDA    #010h
    STA    External_IER     ; Activation des interruptions sur K04-K07
    ANDCCR #00111111b       ; Clear flag i1 and i0 (accept all interrupts)
    RTS

; Interruption du clavier
Int_K04_K07:
    PHALE          ; Sauvegarde de tous les registres
    LDA    K0_Data  ; Lire la valeur du clavier
    ANDA    #00011111b
    STA    clavier_entree ; Ecrire la valeur en RAM
;
; Debug
    ADDA    #030h      ; Transformer la valeur lue en ASCII
    STA    LCD_Data     ; Ecrire le caractere sur l'ecran LCD
;
;           ; Attention les valeurs superieures a 9
;           ; donnent (':' ',' 'B' 'D' 'F' 'H')
;           ; ('*' '#' UP LT RT DW )

```

```
; Debug (oscillation de R50)
  LDA  PortR5_Data
  XORA  #001h          ; Modifier R50 seulement
  STA  PortR5_Data     ; Oscillation du port R50
;
Clavier_pause:
  LDA  K0_Data
  CMPA #01100000b
  BLT  Clavier_pause   ; Attendre que la touche soit relachee
  LDA  #010h
  STA  External_IFR    ; Clear K04_K07 IFF (bit 4)
  PLALE                ; Recuperation de tous les registres
  RTI
;
; Fin de "CLAVIER.ASM"
;-----
```

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:          Pegasus (Stimulateur externe)
-----
; Fichier:         "COMMANDE.ASM"
;
; Description:
;   Gestion des commandes et des menus.
;   Comprend les sous-routines
;   Commande
;
;
; Date:           4 juillet 2002
;
; Auteur:         Stephane Boyer
-----
;
; *** Variables ou registres utilises ***
; fenetre_active   ; Fenetre de reference pour l'interpretation
; clavier_entree   ; Touche entree au clavier
; tmp0             ; Nouvelle valeur (Least Significant Digit)
; tmp1             ; Nouvelle valeur
; tmp2             ; Nouvelle valeur
; tmp3             ; Nouvelle valeur (Most Significant Digit)
;
-----
; Commande de la fenetre principale
Com_Main:
    LDA    clavier_entree
    CMPA   #01h
    BEQ    Com_Main1
    CMPA   #02h
    BEQ    Com_Main2
    CMPA   #03h
    BEQ    Com_Main3
    RTS
Com_Main1:
    LDX    #Aff_OE
    LDY    #Var_OE
    LDA    #01Ah
    STA    fenetre_active    ; Allez a la fenetre parametres
    RTS
Com_Main2:
    LDX    #Aff_Mem
    LDY    #Var_Mem
    LDA    #020h
    STA    fenetre_active    ; Allez a la fenetre de gestion des memoires
    RTS
Com_Main3:
    LDA    #3
    STA    tmp1
    LDA    #0
    STA    tmp0
    LDX    #Aff_StimDuree
    LDY    #Var_StimDuree
    LDA    #030h
    STA    fenetre_active    ; Allez a la fenetre duree de stimulation
    RTS
-----

```

```

; Fenetre parametres (debute a Com_OE)
Com_ParHaut:
    LDX #Aff_OE
    LDY #Var_OE
    LDA #01Ah
    STA fenetre_active      ; Allez a la fenetre parametre OE
    RTS
Com_ParBas:
    LDX #Aff_OI
    LDY #Var_OI
    LDA #01Bh
    STA fenetre_active      ; Allez a la fenetre parametre OI
    RTS
Com_Par1:
    LDX #Aff_OEFRQ
    LDY #Var_OEFRQ
    LDA #011h
    STA fenetre_active      ; Allez a la fenetre parametre OEFRQ
    RTS
Com_Par2:
    LDX #Aff_OEAMP
    LDY #Var_OEAMP
    LDA #012h
    STA fenetre_active      ; Allez a la fenetre parametre OEAMP
    RTS
Com_Par3:
    LDX #Aff_OEPWD
    LDY #Var_OEPWD
    LDA #013h
    STA fenetre_active      ; Allez a la fenetre parametre OEPWD
    RTS
Com_Par4:
    LDX #Aff_OEIPD
    LDY #Var_OEIPD
    LDA #014h
    STA fenetre_active      ; Allez a la fenetre parametre OEIPD
    RTS
Com_Par5:
    LDX #Aff_OIFRQ
    LDY #Var_OIFRQ
    LDA #015h
    STA fenetre_active      ; Allez a la fenetre parametre OIFRQ
    RTS
Com_Par6:
    LDX #Aff_OIAMP
    LDY #Var_OIAMP
    LDA #016h
    STA fenetre_active      ; Allez a la fenetre parametre OIAMP
    RTS
Com_Par7:
    LDX #Aff_OIPWD
    LDY #Var_OIPWD
    LDA #017h
    STA fenetre_active      ; Allez a la fenetre parametre OIPWD
    RTS
Com_Par8:
    LDX #Aff_OIIPD
    LDY #Var_OIIPD
    LDA #018h
    STA fenetre_active      ; Allez a la fenetre parametre OIIPD
    RTS
; -----
; Debut de la sous-routine pour la fenetre 'parametres'

```

; Les 'label' Com_ParX sont plus haut pour eviter les erreurs de phase
;

Com_OE: ; Commande de la fenetre des parametres

Com_OI:

```

    LDA    #020h
    STA    tmp0
    STA    tmp1
    STA    tmp2
    STA    tmp3                ; Init. (espace) des variables temporaires
    LDA    clavier_entree
    CMPA   #012h                ; Fleche vers le haut
    BEQ    Com_ParHaut
    CMPA   #018h                ; Fleche vers le bas
    BEQ    Com_ParBas
    CMPA   #01h
    BEQ    Com_Par1
    CMPA   #02h
    BEQ    Com_Par2
    CMPA   #03h
    BEQ    Com_Par3
    CMPA   #04h
    BEQ    Com_Par4
    CMPA   #05h
    BEQ    Com_Par5
    CMPA   #06h
    BEQ    Com_Par6
    CMPA   #07h
    BEQ    Com_Par7
    CMPA   #08h
    BEQ    Com_Par8
    CMPA   #0Ah                ; Etoile (Back)
    BNE    Com_Sortie
    LDX    #Aff_Main
    LDY    #Var_Main
    LDA    #001h
    STA    fenetre_active      ; Retour a la fenetre principale
    RTS

```

; Commande de modification d'un parametre OE

Com_OEFRQ:

Com_OEAMP:

Com_OEPWD:

Com_OEIPD:

```

    LDA    clavier_entree
    CMPA   #0Bh                ; Carre (Enter)
    BEQ    Com_OEPar1
    CMPA   #0Ah                ; Etoile (Back)
    BEQ    Com_OEParBack
    BGT    Com_OESortie
    LDB    tmp2
    STB    tmp3
    LDB    tmp1
    STB    tmp2
    LDB    tmp0
    STB    tmp1
    STA    tmp0                ; Decalage vers la gauche des 4 caracteres
    RTS

```

Com_OEPar1:

BSR Com_VerParam ; Verification de la valeur du parametre

LDA tmpvalid

LDA #0 ; DEBUG - TBR

BNE Com_OEParBack ; Modification valide - fenetre par. OE

```

    LDA   fenetre_active
    STA   fenetre_old
    LDX   #Aff_ErrNouvParam
    LDA   #0A1h
    STA   fenetre_active      ; Allez a la fenetre erreur parametre
    RTS

Com_OEParBack:
    LDX   #Aff_OE
    LDY   #Var_OE
    LDA   #01Ah
    STA   fenetre_active      ; Retour a la fenetre choix de parametre OE
Com_OESortie:
    RTS

;-----
Com_OIFRQ:                ; Commande de modification d'un parametre OI
Com_OIAMP:
Com_OIPWD:
Com_OIIPD:
    LDA   clavier_entree
    CMPA  #0Bh                ; Carre (Enter)
    BEQ   Com_OIPar1
    CMPA  #0Ah                ; Etoile (Back)
    BEQ   Com_OIParBack
    BGT   Com_OISortie
    LDB   tmp2
    STB   tmp3
    LDB   tmp1
    STB   tmp2
    LDB   tmp0
    STB   tmp1
    STA   tmp0                ; Decalage vers la gauche
    RTS

Com_OIPar1:
    BSR   Com_VerParam        ; Verification de la valeur du parametre
    LDA   tmpvalid
    BNE   Com_OIParBack      ; Modification valide - fenetre par. OI
    LDX   #Aff_ErrNouvParam
    LDA   #0A1h
    STA   fenetre_active      ; Allez a la fenetre erreur parametre
    RTS

Com_OIParBack:
    LDX   #Aff_OI
    LDY   #Var_OI
    LDA   #01Bh
    STA   fenetre_active      ; Retour a la fenetre choix de parametre OI
Com_OISortie:
    RTS

;-----
Com_Mem:                  ; Commande de la fenetre gestion des memoires
    LDA   clavier_entree
    CMPA  #001h                ; Sauvegarder
    BEQ   Com_Mem1
    CMPA  #002h                ; Charger
    BEQ   Com_Mem2
    CMPA  #0Ah                ; Etoile (Back)
    BNE   Com_MemSortie
    LDX   #Aff_Main
    LDY   #Var_Main
    LDA   #001h
    STA   fenetre_active      ; Retour a la fenetre principale
Com_MemSortie:

```

```

RTS
Com_Mem1:
  LDX  #Aff_MemSauv
  LDY  #Var_MemSauv
  LDA  #021h
  STA  fenetre_active      ; Allez a la fenetre sauvegarde memoire
  RTS
Com_Mem2:
; Rafraichir la valeur des parametres temporaires
  BSR  Mem_Refresh_Tmp
  ;
  LDX  #Aff_MemCharge_OE
  LDY  #Var_MemCharge_OE
  LDA  #02Ah
  STA  fenetre_active      ; Allez a la fenetre charge memoire OE
  RTS
; -----
; Sauvegarde des parametres dans la memoire EEPROM
; Chaque ensemble de parametres utilise 6 x 16 bits
; Utilisation de l'index X pour calculer l'adresse memoire
;
Com_MemSauv:      ; Commande de sauvegarde des parametres
  LDA  clavier_entree
  CMPA #01h
  BEQ  Com_MemSauv1      ; Espace memoire A
  CMPA #02h
  BEQ  Com_MemSauv2      ; Espace memoire B
  CMPA #03h
  BEQ  Com_MemSauv3      ; Espace memoire C
  CMPA #04h
  BEQ  Com_MemSauv4      ; Espace memoire D
  CMPA #0Ah
  ; Etoile (Back)
  BNE  Com_MemSauvSortie ; Si different => rien
  LDX  #Aff_Main
  LDY  #Var_Main
  LDA  #001h
  STA  fenetre_active      ; Retour a la fenetre principale
Com_MemSauvSortie:
  RTS
Com_MemSauv1:
  LDA  #0
  BRA  Com_MemSauvSub
Com_MemSauv2:
  LDA  #6
  BRA  Com_MemSauvSub
Com_MemSauv3:
  LDA  #12
  BRA  Com_MemSauvSub
Com_MemSauv4:
  LDA  #18
  BRA  Com_MemSauvSub
Com_MemSauvSub:
  LDZ  #memadr
  STA  M
;
  LDB  oefrq3
  LDA  oefrq2
  PACK
  STA  memMSB
  LDB  oefrq1
  LDA  oefrq0
  PACK

```

```

STA memLSB
BSR Mem_write
;
LDB oeamp1
LDA oeamp0
PACK
STA memMSB
LDB cepwd2
LDA cepwd1
PACK
STA memLSB
LDZ #memadr
INM ; Incremente l'adresse
BSR Mem_write
;
LDB oepwd0
LDA oeipd2
PACK
STA memMSB
LDB oeipd1
LDA oeipd0
PACK
STA memLSB
LDZ #memadr
INM ; Incremente l'adresse
BSR Mem_write
;
LDB oifrq3
LDA oifrq2
PACK
STA memMSB
LDB oifrq1
LDA oifrq0
PACK
STA memLSB
LDZ #memadr
INM ; Incremente l'adresse
BSR Mem_write
;
LDB oiamp1
LDA oiamp0
PACK
STA memMSB
LDB oipwd2
LDA oipwd1
PACK
STA memLSB
LDZ #memadr
INM ; Incremente l'adresse
BSR Mem_write
;
LDB oipwd0
LDA oiipd2
PACK
STA memMSB
LDB oiipd1
LDA oiipd0
PACK
STA memLSB
LDZ #memadr
INM ; Incremente l'adresse
BSR Mem_write
;

```

```

LDX  #Aff_Main
LDY  #Var_Main
LDA  #001h
STA  fenetre_active      ; Retour a la fenetre principale
RTS

;-----
; Recuperation des parametres dans la memoire sEEPROM
;
Com_MemCharge_OE:
Com_MemCharge_OI:      ; Commande de recuperation des parametres
    LDA  clavier_entree
    CMPA #014h          ; Fleche vers la gauche
    BEQ  Com_MemChargeGauche
    CMPA #016h          ; Fleche vers la droite
    BEQ  Com_MemChargeDroite
    CMPA #0Bh           ; Carre (Enter)
    BEQ  Com_MemChargeCourant
    CMPA #0Ah           ; Etoile (Back)
    BNE  Com_MemChargeNoOp
    BRA  Com_MemChargeSortie
;
Com_MemChargeGauche:
    LDA  memoire
    CMPA #"A"
    BEQ  Com_MemChargeNoOp
    DEC  A
    STA  memoire
    BRA  Com_MemChargeSortie
Com_MemChargeDroite:
    LDA  memoire
    CMPA #"D"
    BEQ  Com_MemChargeNoOp
    INC  A
    STA  memoire

    BRA  Com_MemChargeSortie
Com_MemChargeCourant:
    LDA  oefrq3t
    STA  oefrq3
    LDA  oefrq2t
    STA  oefrq2
    LDA  oefrq1t
    STA  oefrq1
    LDA  oefrq0t
    STA  oefrq0
    LDA  oeamp1t
    STA  oeamp1
    LDA  oeamp0t
    STA  oeamp0
    LDA  oepwd2t
    STA  oepwd2
    LDA  oepwd1t
    STA  oepwd1
    LDA  oepwd0t
    STA  oepwd0
    LDA  oeipd2t
    STA  oeipd2
    LDA  oeipd1t
    STA  oeipd1
    LDA  oeipd0t
    STA  oeipd0
    LDA  oefrq3t

```

```

STA  oifrq3
LDA  oifrq2t
STA  oifrq2
LDA  oifrq1t
STA  oifrq1
LDA  oifrq0t
STA  oifrq0
LDA  oiamp1t
STA  oiamp1
LDA  oiamp0t
STA  oiamp0
LDA  oipwd2t
STA  oipwd2
LDA  oipwd1t
STA  oipwd1
LDA  oipwd0t
STA  oipwd0
LDA  oiipd2t
STA  oiipd2
LDA  oiipd1t
STA  oiipd1
LDA  oiipd0t
STA  oiipd0
;
Com_MemChargeSortie:
LDX  #Aff_Main
LDY  #Var_Main
LDA  #001h
STA  fenetre_active      ; Retour a la fenetre principale
Com_MemChargeNoOP:
RTS

;-----
Com_StimDuree:          ; Commande de duree de stimulation
LDA  clavier_entree
CMPA #0Bh               ; Carre (Enter)
BEQ  Com_StimDuree1
CMPA #0Ah               ; Etoile (Back)
BEQ  Com_StimDureeBack
BGT  Com_StimSortie
LDB  tmp0
STB  tmp1
STA  tmp0               ; Decalage vers la gauche
RTS

Com_StimDuree1:
; Verification de la duree de stimulation
LDA  tmp1               ; Recuperation des dizaines
CMPB #009h
BGT  Com_StimDureeErr   ; Valeur des dizaines non-decimale
TFR  A,B               ; B <= A
SLL  A
SLL  A
SLL  A
ADDA B
ADCA B                 ; Multiplication par 10
; A contient la valeur de tmp1 multiplie par 10
LDB  tmp0               ; Recuperation des unites
CMPB #009h
BGT  Com_StimDureeErr   ; Valeur des unites non-decimale
ADDA B                 ; Conversion BCD -> binaire
; A contient la valeur de 10*tmp1+tmp0
CMPA #5
BLT  Com_StimDureeErr   ; Duree plus petite que 5

```

```

; Duree valide => Sauvegarde la duree
LDA tmp1
STA duree1
LDA tmp0
STA duree0
; Duree valide => Commencer la stimulation
LDA #0F0h
STA clavier_entree ; Stimulation active (clavier = #0F0h)
; BSR Com_ValideAll ; Verifier la coherence des parametres TBC
; Activer l'interruption pour 1 Hz

LDX #Aff_StimActive
LDY #Var_StimActive
LDA #031h
STA fenetre_active ; Allez a la fenetre stimulation active
RTS
Com_StimDureeErr:
LDX #Aff_ErrStimDuree
LDA #0A0h
STA fenetre_active ; Allez a la fenetre erreur de duree de stim.
RTS
Com_StimDureeBack:
LDX #Aff_Main
LDY #Var_Main
LDA #001h
STA fenetre_active ; Retour a la fenetre principale
Com_StimSortie:
RTS

;-----
; Interruption de la stimulation avant l'echec du temps
; Desactiver la stimulation
; *** TBC ***
Com_StimActive: ; Commande stimulation active
LDX #Aff_Main
LDY #Var_Main
LDA #001h
STA fenetre_active ; Retour a la fenetre principale
RTS

;-----
Com_Erreur: ; Commande lors d'une erreur
LDX #Aff_Main
LDY #Var_Main
LDA #001h
STA fenetre_active ; Retour a la fenetre principale
RTS

;-----
Com_ParErreur: ; Commande lors d'une erreur de parametre
LDA #020h
STA tmp0
STA tmp1
STA tmp2
STA tmp3 ; Reinit. (espace) des variables temporaires
LDA fenetre_old
CMPA #011h
BEQ Com_Par1
CMPA #012h
BEQ Com_Par2
CMPA #013h
BEQ Com_Par3

```

```

CMPA #014h
BEQ Com_Par4
CMPA #015h
BEQ Com_Par5
CMPA #016h
BEQ Com_Par6
CMPA #017h
BEQ Com_Par7
BRA Com_Par8 ; Le retour RTS dans <Fenetre parametres>

; Specification d'une adresse de reference pour les erreurs de phase
ORG 0780h

;-----
; La verification des parametres est basee sur le nombre de bits
; pour chacun des parametres dans le generateur de stimuli et sur
; la frequence de l'horloge alimentant ce circuit.
; Pour une horloge de 1 MHz (T = 1.0 us):
;   Amplitude:      5 bits | AMP = 0 -> 2.0 mA | 2 digits
;   Periode:        16 bits | FRQ = 20 -> 2000 Hz | 4 digits
;   Largeur d'une impulsion: 8 bits | PWD = 0 -> 255 us | 3 digits
;   Delai inter-impulsion: 8 bits | IPD = 0 -> 255 us | 3 digits
;
; Hypothese: Les variables tmp[0:3] ne contiennent que des chiffres [0-9]
; ou un espace (020h)
;
Com_VerOEFRQ: ; Verification (Frequence OE)
    CMPD #20
    BLT Com_VerEchec
    CMPD #2000
    BGT Com_VerEchec
    LDA tmp3
    STA oefrq3
    LDA tmp2
    STA oefrq2
    LDA tmp1
    STA oefrq1
    LDA tmp0
    STA oefrq0
    LDA #1
    STA tmpvalid
    RTS
Com_VerOEAMP: ; Verification (Amplitude OE)
    CMPD #20
    BGT Com_VerEchec
    LDA tmp1
    STA oeamp1
    LDA tmp0
    STA oeamp0
    LDA #1
    STA tmpvalid
    RTS
Com_VerOEPWD: ; Verification (Largeur d'impulsion OE)
    CMPD #255
    BGT Com_VerEchec
    LDA tmp2
    STA oepwd2
    LDA tmp1
    STA oepwd1
    LDA tmp0
    STA oepwd0
    LDA #1
    STA tmpvalid

```

```

RTS
Com_VerOEIPD:                ; Verification (Delai inter-impulsion OE)
    CMPD    #255
    BGT     Com_VerEchec
    LDA     tmp2
    STA     oeipd2
    LDA     tmp1
    STA     oeipd1
    LDA     tmp0
    STA     oeipd0
    LDA     #1
    STA     tmpvalid
    RTS

Com_VerOIFRQ:                ; Verification (Frequence OI)
    CMPD    #20
    BLT     Com_VerEchec
    CMPD    #2000
    BGT     Com_VerEchec
    LDA     tmp3
    STA     oifrq3
    LDA     tmp2
    STA     oifrq2
    LDA     tmp1
    STA     oifrq1
    LDA     tmp0
    STA     oifrq0
    LDA     #1
    STA     tmpvalid
    RTS

Com_VerOIAMP:                ; Verification (Amplitude OI)
    CMPD    #20
    BGT     Com_VerEchec
    LDA     tmp1
    STA     oiamp1
    LDA     tmp0
    STA     oiamp0
    LDA     #1
    STA     tmpvalid
    RTS

Com_VerOIPWD:                ; Verification (Largeur d'impulsion OI)
    CMPD    #255
    BGT     Com_VerEchec
    LDA     tmp2
    STA     oipwd2
    LDA     tmp1
    STA     oipwd1
    LDA     tmp0
    STA     oipwd0
    LDA     #1
    STA     tmpvalid
    RTS

Com_VerOIIPD:                ; Verification (Delai inter-impulsion OI)
    CMPD    #255
    BGT     Com_VerEchec
    LDA     tmp2
    STA     oiipd2
    LDA     tmp1
    STA     oiipd1
    LDA     tmp0
    STA     oiipd0
    LDA     #1
    STA     tmpvalid
    RTS

```

```

Com_tmpAllSpace:
    LDA    #OFFh
    STA    tmpvalid
    RTS

Com_VerEchec:
    LDA    #0
    STA    tmpvalid
    RTS

;-----
; Debut
; La fin ci-haut pour les erreurs de phase
;-----

Com_VerParam:
    LDA    fenetre_active
    ANDA    #0F3h
    CMPA    #011h                ; Frequence
    BEQ    Com_4digit
    CMPA    #013h                ; Largeur d'impulsion
    BEQ    Com_3digit
    CMPA    #014h                ; Delai inter-impulsion
    BEQ    Com_3digit
    LDA    #" "                  ; Amplitude (012h)
    STA    tmp2

Com_3digit:
    STA    tmp3

Com_4digit:                ; Calcul de la valeur total
    LDH    #0
    LDL    tmp3
    CMPL    #" "
    BEQ    Com_tmp3isSpace
    CMPL    #039h
    BLT    Com_VerEchec
    ANDL    #0Fh
    TFR    H,B                ; MSB
    TFR    L,A                ; LSB

Com_tmp3isSpace:
    LDL    tmp2                ; H = 0
    CMPL    #" "
    BEQ    Com_tmp2isSpace
    CMPL    #039h
    BLT    Com_VerEchec
    ANDL    #0Fh

;
    TFR    D,X                ; IX <- BA
    ADDD    D                ; 2
    ADDD    D                ; 4
    ADDD    D                ; 8
    ADDD    X                ; 9
    ADDD    X                ; 10
    ADDD    Z                ; BA <- (BA * 10) + HL

Com_tmp2isSpace:
    LDL    tmp1                ; H = 0
    CMPL    #" "
    BEQ    Com_tmp1isSpace
    CMPL    #039h
    BLT    Com_VerEchec
    ANDL    #0Fh

;
    TFR    D,X                ; IX <- BA
    ADDD    D                ; 2
    ADDD    D                ; 4
    ADDD    D                ; 8
    ADDD    X                ; 9

```

```

        ADDD X                ; 10
        ADDD Z                ; BA <- (BA * 10) + HL
Com_tmplisSpace:
        LDL tmp0              ; H = 0
        CMPL #" "
        BEQ Com_tmpAllSpace
        CMPL #039h
        BLT Com_VerEchec
        ANDL #0Fh
;
        TFR D,X               ; IX <- BA
        ADDD D                ; 2
        ADDD D                ; 4
        ADDD D                ; 8
        ADDD X                ; 9
        ADDD X                ; 10
        ADDD Z                ; BA <- (BA * 10) + HL
                                ; BA: Valeur total sur 16 bits
; Identifier la comparaison a faire
        LDL fenetre_active
        CMPL #011h            ; Frequence OE
        BEQ Com_VerOEFRQ
        CMPL #012h            ; Amplitude OE
        BEQ Com_VerOEAMP
        CMPL #013h            ; Largeur d'impulsion OE
        BEQ Com_VerOEPWD
        CMPL #014h            ; Delai inter-impulsion OE
        BEQ Com_VerOEIPD
        CMPL #015h            ; Frequence OI
        BEQ Com_VerOIFRQ
        CMPL #016h            ; Amplitude OI
        BEQ Com_VerOIAMP
        CMPL #017h            ; Largeur d'impulsion OI
        BEQ Com_VerOIPWD
        BRA Com_VerOIIPD      ; Delai inter-impulsion OI
; Suite au debut de la routine pour les erreurs de phase

;-----
; Interpretation de la commande
Commande:
        LDA fenetre_active    ; Selon la fenetre active, interpreter l'entree
                                ; au clavier dans la sous-routine appropriee
        CMPA #001h
        BEQ Com_Main
        CMPA #01Ah
        BEQ Com_OE
        CMPA #01Bh
        BEQ Com_OI
        CMPA #011h
        BEQ Com_OEFRQ
        CMPA #012h
        BEQ Com_OEAMP
        CMPA #013h
        BEQ Com_OEPWD
        CMPA #014h
        BEQ Com_OEIPD
        CMPA #015h
        BEQ Com_OIFRQ
        CMPA #016h
        BEQ Com_OIAMP
        CMPA #017h
        BEQ Com_OIPWD
        CMPA #018h

```

```

BEQ  Com_OIIPD
CMPA #020h
BEQ  Com_Mem
CMPA #021h
BEQ  Com_MemSauv
CMPA #022h
BEQ  Com_MemCharge_OE
CMPA #023h
BEQ  Com_MemCharge_OI
CMPA #030h
BEQ  Com_StimDuree
CMPA #031h
BEQ  Com_StimActive
CMPA #0A0h
BEQ  Com_Erreur
CMPA #0A1h
BEQ  Com_ParErreur
CMPA #0A2h
BEQ  Com_Erreur
; No de fenetre errone
; Afficher la variable fenetre_active
; sur le LCD en BCD (attention avec A,B,C,D,E,F)
LDB  #80h
STB  LCD_Command      ; Deplacer le curseur sur la premiere ligne
LDZ  LCD_Data
LDM  #"F"
LDM  #"e"
LDM  #"n"
LDM  #". "
LDM  #" "
LDM  #"i"
LDM  #"n"
LDM  #"c"
LDM  #"o"
LDM  #"n"
LDM  #"n"
LDM  #"u"
LDM  #"e"
LDM  #" "
LDA  fenetre_active
UPCK                               ; MSB dans B, LSB dans A
CMPA #00Ah
BLT  Com_NoHexMSB
ADDA #007h
Com_NoHexMSB:
ADDA #030h
STA  LCD_Data      ; Affiche les MSB en BCD
LDA  #030h
ADDA B
CMPB #00Ah
BLT  Com_NoHexLSB
ADDA #007h
Com_NoHexLSB:
STA  LCD_Data      ; Affiche les LSB en BCD
SLP                               ; Sleep mode
Com_Sortie:
RTS
;
; Fin de "COMMANDE.ASM"
;-----

```

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:          Pegasus (Stimulateur externe)
-----
; Fichier:         "MEMOIRE.ASM"
;
; Description:
;   Gestion de l'initialisation, de la sauvegarde et de
;   la recuperation des parametres de stimulation
;   Comprend les routine:
;       Par_Init
;       . Charge en memoire les valeurs de parametres par defaut
;
; Date:           3 juillet 2002
;
; Auteur:         Stephane Boyer
-----
;
; *** Variables utilisees ***
;
-----
; Initialisation des parametres
Mem_Init:
    LDA    #"A"
    STA    memoire           ; Espace memoire: sEEPROM (A,B,C,D)
    LDA    #" "
    STA    oefrq3
    LDA    #" "
    STA    oefrq2
    LDA    #3
    STA    oefrq1
    LDA    #0
    STA    oefrq0           ; Frequence de l'onde excitante (OE) 0030 Hz
    LDA    #1
    STA    oeamp1
    LDA    #2
    STA    oeamp0           ; Amplitude de l'OE 1.2 mA
    LDA    #3
    STA    oepwd2
    LDA    #0
    STA    oepwd1
    LDA    #0
    STA    oepwd0           ; Largeur d'impulsion de l'OE 300 us
    LDA    #" "
    STA    oeipd2
    LDA    #" "
    STA    oeipd1
    LDA    #0
    STA    oeipd0           ; Delai inter-impulsion de l'OE 000 us
    LDA    #" "
    STA    oifrq3
    LDA    #6
    STA    oifrq2
    LDA    #0
    STA    oifrq1
    LDA    #0
    STA    oifrq0           ; Frequence de l'onde inhibitrice (OI) 600 Hz
    LDA    #0
    STA    oiamp1
    LDA    #9

```

```

STA oiamp0          ; Amplitude de l'OI 0.9 mA
LDA #" "
STA oipwd2
LDA #6
STA oipwd1
LDA #0
STA oipwd0          ; Largeur d'impulsion de l'OI 060 us
LDA #" "
STA oiipd2
LDA #" "
STA oiipd1
LDA #0
STA oiipd0          ; Delai inter-impulsion de l'OI 000 us
RTS
;
;-----
; Ecriture de donnees en memoire
; Adresse: memadr
; donnees: memMSB, memLSB
; sEPROM - P1 chart (GO, SHIFT_EN, CS, PE, PRE, SCLK, DI, DO)
; Note: Le PortP1 doit être en sortie sauf le bit 0 (D0)
Mem_write:
;-----
; Write enable (00 11XXXX)
;-----
LDA #00101000b      ; CS <= '1' (P15)
STA PortP1_Data
LDA #00101010b      ; DI <= '1' (start bit)
STA PortP1_Data
LDA #00101110b      ; Clock rising edge
STA PortP1_Data
LDA #00101000b      ; DI <= '0'
STA PortP1_Data
LDA #00101100b      ; Clock rising edge
STA PortP1_Data
LDA #00101000b      ; DI <= '0'
STA PortP1_Data
LDA #00101100b      ; Clock rising edge
STA PortP1_Data
LDA #00101010b      ; DI <= '1'
STA PortP1_Data
LDA #00101110b      ; Clock rising edge
STA PortP1_Data
LDA #00101010b      ; DI <= '1'
STA PortP1_Data
LDA #00101110b      ; Clock rising edge
STA PortP1_Data
LDA #00101000b      ; DI <= 'X'
STA PortP1_Data
LDA #00101100b      ; Clock rising edge
STA PortP1_Data
LDA #00101000b      ; DI <= 'X'
STA PortP1_Data
LDA #00101100b      ; Clock rising edge
STA PortP1_Data
LDA #00101000b      ; DI <= 'X'
STA PortP1_Data
LDA #00101100b      ; Clock rising edge
STA PortP1_Data
LDA #00101000b      ; DI <= 'X'
STA PortP1_Data
LDA #00101100b      ; Clock rising edge
STA PortP1_Data

```

```

LDA #00101000b      ; Clock <= '0'
STA PortP1_Data
LDA #00001000b      ; CS <= '0'
STA PortP1_Data
;
;-----
; Write instruction (01 A5:A0 D15:D0)
;-----
LDA #00101000b      ; CS <= '1' (P15)
STA PortP1_Data
LDA #00101010b      ; DI <= '1' (start bit)
STA PortP1_Data
LDA #00101110b      ; Clock rising edge
STA PortP1_Data
LDA #00101000b      ; DI <= '0'
STA PortP1_Data
LDA #00101100b      ; Clock rising edge
STA PortP1_Data
LDA #00101010b      ; DI <= '1'
STA PortP1_Data
LDA #00101110b      ; Clock rising edge
STA PortP1_Data
;
LDY #3               ; Nombre d'octets a envoyer
LDB memadr           ; Charger l'adresse
Mem_WR_Reload:
  CMPY #3
  BEQ Mem_WR_Skip
  LDB memMSB          ; Charger les MSB
  CMPY #2
  BEQ Mem_WR_MSB
  LDB memLSB
Mem_WR_MSB:
  LDX #8              ; Charger le nombre de bits
  BRA Mem_WR_Transmit
Mem_WR_Skip:
  LDX #6              ; Charger le nombre de bits
  SLL B               ; C <- DDDDDDDD <- 0
  SLL B               ; C <- DDDDDDDD <- 0
Mem_WR_Transmit:
  DEC X               ; Decrementer X (compteur de bits)
  SLL B               ; C <- DDDDDDDD <- 0
  BCC Mem_WR_Write0   ; Branch on Carry Clear
Mem_WR_Write1:
  LDA #00101010b      ; DI <= '1'
  STA PortP1_Data
  LDA #00101110b      ; Clock rising edge
  STA PortP1_Data
  BRA Mem_WR_Test
Mem_WR_Write0:
  LDA #00101000b      ; DI <= '0'
  STA PortP1_Data
  LDA #00101100b      ; Clock rising edge
  STA PortP1_Data
Mem_WR_Test:
  CMPX #0
  BNE Mem_WR_Transmit ; S'il reste des bits a envoyer
;
DEC Y                ; Decrementer le compteur d'octets
CMPY #0
BNE Mem_WR_Reload
;
LDA #00101000b      ; Clock <= '0'

```

```

STA  PortP1_Data
LDA  #00001000b      ; CS  <= '0'
STA  PortP1_Data
;
;-----
; Write disable (00 00XXXX)
;-----
LDA  #00101000b      ; CS <= '1' (P15)
STA  PortP1_Data
LDA  #00101010b      ; DI <= '1' (start bit)
STA  PortP1_Data
LDA  #00101110b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= '0'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= '0'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= '0'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= '0'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= 'X'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= 'X'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= 'X'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; DI <= 'X'
STA  PortP1_Data
LDA  #00101100b      ; Clock rising edge
STA  PortP1_Data
LDA  #00101000b      ; Clock <= '0'
STA  PortP1_Data
LDA  #00001000b      ; CS  <= '0'
STA  PortP1_Data
RTS
;
;-----
; Lecture de donnees de la memoire
; Adresse: memadr
; Return: memMSB, memLSB
; sEPROM - P1 chart (GO, SHIFT_EN, CS, PE, PRE, SCLK, DI, D0)
; Note:  Le PortP1 doit être en sortie, sauf le bit 0 (D0)
Mem_read:
;-----
; Read instruction (10 A5:A0)
;-----
LDA  #00101000b      ; CS <= '1' (P15)
STA  PortP1_Data

```

```

LDA #00101010b ; DI <= '1' (start bit)
STA PortP1_Data
LDA #00101110b ; Clock rising edge
STA PortP1_Data
LDA #00101010b ; DI <= '1'
STA PortP1_Data
LDA #00101110b ; Clock rising edge
STA PortP1_Data
LDA #00101000b ; DI <= '0'
STA PortP1_Data
LDA #00101100b ; Clock rising edge
STA PortP1_Data
;
LDB memadr ; Charger l'adresse
LDX #6 ; Nombre de bits a envoyer pour l'adresse
SLL B ; C <- DDDDDDDD <- 0
SLL B ; C <- DDDDDDDD <- 0
Mem_RD_Transmit:
DEC X ; Decremente X (compteur de bits)
SLL B ; C <- DDDDDDDD <- 0
BCC Mem_RD_Write0 ; Branch on carry clear
Mem_RD_Write1:
LDA #00101010b ; DI <= '1'
STA PortP1_Data
LDA #00101110b ; Clock rising edge
STA PortP1_Data
BRA Mem_RD_Test
Mem_RD_Write0:
LDA #00101000b ; DI <= '0'
STA PortP1_Data
LDA #00101100b ; Clock rising edge
STA PortP1_Data
Mem_RD_Test:
CMPX #0
BNE Mem_RD_Transmit ; S'il reste des bits a envoyer
;
; Lecture des donnees en commençant par D15
LDX #16 ; Nombre de bits a lire
LDB #0h ; Donnee
Mem_RD_Read:
LDA #00101000b ; DI <= '0', D0
STA PortP1_Data
LDA #00101100b ; Clock rising edge
STA PortP1_Data
DEC X ; Decremente X (compteur de bits)
SLA B ; Shift Left C <- DDDDDDDD <- 0
LDA PortP1_Data ; Read D0
ANDX #01h
BEQ Mem_RD_D0is0
; D0 is 1
ORB #01h ; Set LSB to 1
Mem_RD_D0is0:
CMPX #8
BEQ Mem_RD_StoreMSB ; Lecture des MSB terminee
CMPX #0
BEQ Mem_RD_StoreLSB ; Lecture des LSB terminee
BRA Mem_RD_Read
;
Mem_RD_StoreMSB:
STB memMSB
LDB #0
BRA Mem_RD_read

```

```

Mem_RD_StoreLSB:
    STB    memLSB
;
    LDA    #00101000b        ; Clock <= '0'
    STA    PortP1_Data
    LDA    #00001000b        ; CS    <= '0'
    STA    PortP1_Data
    RTS

;-----
; Rafraichir la valeur des parametres temporaires
; Fonction de la valeur de 'memoire' (A, B, C, D)
;
Mem_Refresh_Tmp:
    LDA    memoire
    SUBA    #"A"
    SLL     A
    TFR     A,B                ; B <= A
    SLL     A
    ADDA    B                  ; A = 6 * (memoire - "A")
    LDZ     #memadr
;
    STA     M
    BSR     Mem_read
    LDA     memMSB
    UPCK
    STB     oefrq3t
    STA     oefrq2t
    LDA     memLSB
    UPCK
    STB     oefrq1t
    STA     oefrq0t
;
    LDZ     #memadr
    INM
    BSR     Mem_read
    LDA     memMSB
    UPCK
    STB     oeamp1t
    STA     oeamp0t
    LDA     memLSB
    UPCK
    STB     oepwd2t
    STA     oepwd1t
;
    LDZ     #memadr
    INM
    BSR     Mem_read
    LDA     memMSB
    UPCK
    STB     oeipd1t
    STA     oeipd0t
;
    LDZ     #memadr
    INM
    BSR     Mem_read
    LDA     memMSB
    UPCK
    STB     oifrq3t

```

```

    STA    oifrq2t
    LDA    memLSB
    UPCK
    STB    oifrq1t
    STA    oifrq0t
;
    LDZ    #memadr
    INM
    BSR    Mem_read
    LDA    memMSB
    UPCK
    STB    oiamp1t
    STA    oiamp0t
    LDA    memLSB
    UPCK
    STB    oipwd2t
    STA    oipwd1t
;
    LDZ    #memadr
    INM
    BSR    Mem_read
    LDA    memMSB
    UPCK
    STB    oipwd0t
    STA    oiipd2t
    LDA    memLSB
    UPCK
    STB    oiipd1t
    STA    oiipd0t
;
    RTS                                     ; Retour de Mem_Refresh_Tmp

;-----
; Decremente la duree de stimulation
; Termine la stimulation si la duree = 0
;
Int_01Hz:
    LDB    duree1
    LDA    duree0
    PACK
    BNE    Int_01Hz_Dec
    ; Fin de stimulation
    LDX    #Aff_Main
    LDY    #Var_Main
    LDA    #001h
    STA    fenetre_active                 ; Retour a la fenetre principale
    RTS
Int_01Hz_Dec:
    ORCCR #010h                           ; Operation decimal
    DEC    A
    UPCK
    STB    duree1
    STA    duree0
    ANDCCR #0EFh                           ; Operation binaire
    RTS
;
; Fin de "MEMOIRE.ASM"
;-----

```

```

-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet:          Pegasus (Stimulateur externe)
-----
; Fichier:         "LCD.ASM"
;
; Description:
;   Gestion de l'affichage sur l'ecran LCD.
;   Comprend les sous-routines
;       LCD_Init
;           . Initialise le LCD
;           . Cree les nouveaux caracteres (fleches) dans la CGRAM du LCD
;
;       LCD_Update
;           . Affiche la fenetre dont le 1er caractere est pointe par IX
;           . Gere l'affichage des variables
;
; Date:           3 juillet 2002
;
; Auteur:         Stephane Boyer
-----
;
; *** Variables ou registres utilises ***
; IX              ; Adresse du 1er caractere a afficher
; IY              ; Adresse du pointeur du 1er caracteres
;
-----
; Initialisation LCD
LCD_Init:
    LDA    #0Eh
    STA    LCD_Command      ; Display ON, cursor ON, not blinking
    LDA    #38h
    STA    LCD_Command      ; 8 bits, 4 lines, 5x7 dots
    LDA    #01h
    STA    LCD_Command      ; Clear display
LCD_Busy:
    LDA    LCD_Command      ; Read LCD buzy flag (Buzy: DB7=1)
    ANDA   #80h             ; If buzy, A <= 80h
    BNE    LCD_Busy         ; Branch if display busy (Z=0)
; Nouveaux caracteres
; Fleche haut CG RAM (6)
; Fleche bas  CD RAM (7)
    LDA    #070h
    STA    LCD_Command      ; Set CG RAM address(6)
    LDZ    #LCD_Data
    LDM    #00h              ; Write to CG RAM (6)
    LDM    #004h
    LDM    #00Eh
    LDM    #015h
    LDM    #004h
    LDM    #004h
    LDM    #004h
    LDM    #004h
    LDM    #00h
    LDM    #00h              ; Write to CG RAM (7)
    LDM    #004h
    LDM    #004h
    LDM    #004h
    LDM    #015h
    LDM    #00Eh
    LDM    #004h

```

```

LDM    #00h
RTS

;
;-----
; Affichage sur l'ecran LCD
; 4 lignes de 16 caracteres
;
LCD_Update:
LDH    #0F0h           ; MSB de l'adresse des variables dans la RAM
LDA    #080h
STA    LCD_Command     ; Positionner le curseur sur la premiere ligne
LDL    #00h            ; Nombre de caracteres affiches

LCD_prochain_caractere:
LDA    L,X             ; A <- [IX+L]
CMPA   #"$"           ; Variable
BNE    LCD_aff_caractere
TFR    L,B             ; Sauvegarde de la position (tmp <= position)
LDL    L,Y             ; LSB de l'adresse de la variable dans la RAM
LDA    M               ; Charge la valeur binaire

CMPA   #00Ah          ; BCD si plus petit que 0Ah
BGE    **4             ; Deja un caractere ASCII
ADDA   #030h          ; Transformer d'un chiffre binaire en ASCII
TFR    B,L            ; Retour de la position (position <= tmp)

LCD_aff_caractere:
STA    LCD_Data        ; Ecrire le caractere sur le LCD
INC    L               ; Incrmente le nombre de caracteres affichees
CPL    #16            ; Verification de position (fin de ligne)
BEQ    LCD_Ligne2      ; Positionner le curseur sur la deuxieme ligne
CPL    #32
BEQ    LCD_Ligne3      ; Positionner le curseur sur la troisieme ligne
CPL    #48
BEQ    LCD_Ligne4      ; Positionner le curseur sur la quatrieme ligne
CPL    #64
BLT    LCD_prochain_caractere
RTS

;
LCD_Ligne2:
LDA    #0C0h
STA    LCD_Command
BRA    LCD_prochain_caractere

LCD_Ligne3:
LDA    #090h
STA    LCD_Command
BRA    LCD_prochain_caractere

LCD_Ligne4:
LDA    #0D0h
STA    LCD_Command
BRA    LCD_prochain_caractere

;
; Fin de "LCD.ASM"
;-----

```

```

;-----
; PolySTIM
; Ecole Polytechnique de Montreal
;
; Projet: Stimulateur externe
;-----
; Fichier: "LCD_TBL.ASM"
;
; Description:
; Tableau du texte a afficher sur l'ecran LCD.
; Organise en 20 fenetres de 64 caracteres.
; Debut des tableaux aux signets
; Aff_Table
; Var_Table
;
; Date: 3 juillet 2002
;
; Auteur: Stephane Boyer
;-----
;
Aff_Table: DFS {20*64} ; 20 fenetres de 16x4 caracteres
ORG Aff_Table
;
Aff_Init: ; Fenetre 00h
    DFB "*****"
    DFB " Initialisation "
    DFB " en cours "
    DFB "*****"
;
Aff_Main: ; Fenetre 001h
    DFB "Menu principal "
    DFB "1-Parametres "
    DFB "2-Memoires "
    DFB "3-Stimulation "
;
Aff_OE: ; Fenetre 01Ah
    DFB "OE 1-FRQ $$$$ Hz"
    DFB " 2-AMP $. $ mA"
    DFB 00Fh ; F_bas
    DFB " 3-PWD $$$ us"
    DFB " 4-IPD $$$ us"
;
Aff_OI: ; Fenetre 01Bh
    DFB "OI 5-FRQ $$$$ Hz"
    DFB " 6-AMP $. $ mA"
    DFB 00Eh ; F_haut
    DFB " 7-PWD $$$ us"
    DFB " 8-IPD $$$ us"
;
Aff_OEFRQ: ; Fenetre 011h
    DFB "Onde excitante "
    DFB " FRQ $$$$ Hz "
    DFB "Nouvelle valeur "
    DFB " $$$$ Hz "
;
Aff_OEAMP: ; Fenetre 012h
    DFB "Onde excitante "
    DFB " AMP $. $ mA "
    DFB "Nouvelle valeur "
    DFB " $. $ mA "
;
Aff_OEPWD: ; Fenetre 013h

```

```

DFB      "Onde excitante "
DFB      "   PWD   $$$ us "
DFB      "Nouvelle valeur "
DFB      "       $$$ us "
;
Aff_OEIPD:                                ; Fenetre 014h
DFB      "Onde excitante "
DFB      "   IPD   $$$ us "
DFB      "Nouvelle valeur "
DFB      "       $$$ us "
;
Aff_OIFRQ:                                ; Fenetre 015h
DFB      "Onde inhibitrice"
DFB      "   FRQ   $$$$ Hz "
DFB      "Nouvelle valeur "
DFB      "       $$$$ Hz "
;
Aff_OIAMP:                                ; Fenetre 016h
DFB      "Onde inhibitrice"
DFB      "   AMP   $. $ mA "
DFB      "Nouvelle valeur "
DFB      "       $. $ mA "
;
Aff_OIPWD:                                ; Fenetre 017h
DFB      "Onde inhibitrice"
DFB      "   PWD   $$$ us "
DFB      "Nouvelle valeur "
DFB      "       $$$ us "
;
Aff_OIIPD:                                ; Fenetre 018h
DFB      "Onde inhibitrice"
DFB      "   IPD   $$$ us "
DFB      "Nouvelle valeur "
DFB      "       $$$ us "
;
Aff_Mem:                                  ; Fenetre 020h
DFB      "Memoires      "
DFB      "1-Sauvegarder "
DFB      "2-Charger     "
DFB      "              "
;
Aff_MemSauv:                              ; Fenetre 021h
DFB      "Sauvegarder dans"
DFB      "la memoire     "
DFB      "1- A      3- C  "
DFB      "2- B      4- D  "
;
Aff_MemCharge_OE:                          ; Fenetre 02Ah
DFB      "$) FRQ $$$$ Hz "
DFB      "OE AMP $. $ mA "
DFB      00Fh                                ; F_bas
DFB      "   PWD   $$$ us "
DFB      07Fh,07Eh                          ; F_gauche, F_droite
DFB      "   IPD   $$$ us "
;
Aff_MemCharge_OI:                          ; Fenetre 02Bh
DFB      "$) FRQ $$$$ Hz "
DFB      "OI AMP $. $ mA "
DFB      00Eh                                ; F_haut
DFB      "   PWD   $$$ us "
DFB      07Fh,07Eh                          ; F_gauche, F_droite
DFB      "   IPD   $$$ us "
;

```

```

Aff_StimDuree:                                ; Fenetre 030h
    DFB "Duree de "
    DFB "stimulation: "
    DFB "  $$ secondes " ; Par default (30s)
    DFB "
;
Aff_StimActive:                                ; Fenetre 031h
    DFB "* STIM. ACTIVE *"
    DFB "  $$ secondes " ; Compte a rebours
    DFB "[Une touche "
    DFB " pour arreter] "
;
Aff_ErrStimDuree:                              ; Fenetre 0A0h
    DFB "**** ERREUR ****"
    DFB " La duree doit "
    DFB " etre entre 05 "
    DFB " et 99 secondes."
;
Aff_ErrNouvParam:                             ; Fenetre 0A1h
    DFB "**** ERREUR ****"
    DFB " Nouvelle valeur"
    DFB " en dehors de la"
    DFB " plage permise. "
;
Aff_ErrPar:                                    ; Fenetre 0A2h
    DFB "**** ERREUR ****"
    DFB "Stim. impossible"
    DFB " Verifier les "
    DFB " parametres. "
;
-----
; Tableau d'index pour les variables
; Partie basse des adresses (11)
;
;
Var_table:  DFS    {20*64}    ; 20 fen. de 16x4 car.
ORG Var_table
;
Var_Init:   DFS    {64}       ; Pas de variables
;
Var_Main:   DFS    {64}       ; Pas de variables
;
Var_OE:
    DFB "OE 1-FRQ "
    DFB LOW{oefrq3}
    DFB LOW{oefrq2}
    DFB LOW{oefrq1}
    DFB LOW{oefrq0}
    DFB " Hz"
;
    DFB " 2-AMP "
    DFB LOW{oeamp1}
    DFB " "
    DFB LOW{oeamp0}
    DFB " mA"
;
    DFB "* 3-PWD "
    DFB LOW{oepwd2}
    DFB LOW{oepwd1}
    DFB LOW{oepwd0}
    DFB " us"
;
    DFB " 4-IPD "

```

```

        DFB      LOW{oeipd2}
        DFB      LOW{oeipd1}
        DFB      LOW{oeipd0}
        DFB      " us"
;
Var_OI:
        DFB      "OI 5-FRQ "
        DFB      LOW{oifrq3}
        DFB      LOW{oifrq2}
        DFB      LOW{oifrq1}
        DFB      LOW{oifrq0}
        DFB      " Hz"
;
        DFB      " 6-AMP "
        DFB      LOW{oiamp1}
        DFB      "."
        DFB      LOW{oiamp0}
        DFB      " mA"
;
        DFB      "* 7-PWD "
        DFB      LOW{oipwd2}
        DFB      LOW{oipwd1}
        DFB      LOW{oipwd0}
        DFB      " us"
;
        DFB      " 8-IPD "
        DFB      LOW{oiipd2}
        DFB      LOW{oiipd1}
        DFB      LOW{oiipd0}
        DFB      " us"
;
Var_OEFRQ:
        DFB      "Onde excitante "
;
        DFB      " FRQ "
        DFB      LOW{oefrq3}
        DFB      LOW{oefrq2}
        DFB      LOW{oefrq1}
        DFB      LOW{oefrq0}
        DFB      " Hz "
;
        DFB      "Nouvelle valeur "
;
        DFB      " "
        DFB      LOW{tmp3}
        DFB      LOW{tmp2}
        DFB      LOW{tmp1}
        DFB      LOW{tmp0}
        DFB      " Hz "
;
Var_OEAMP:
        DFB      "Onde excitante "
;
        DFB      " AMP "
        DFB      LOW{oeamp1}
        DFB      "."
        DFB      LOW{oeamp0}
        DFB      " mA "
;
        DFB      "Nouvelle valeur "
;
        DFB      " "
        DFB      LOW{tmp1}

```

```

DFB      " "
DFB      LOW{tmp0}
DFB      " mA "
;
Var_OEPWD:
DFB      "Onde excitante "
;
DFB      " PWD "
DFB      LOW{oepwd2}
DFB      LOW{oepwd1}
DFB      LOW{oepwd0}
DFB      " us "
;
DFB      "Nouvelle valeur "
;
DFB      " "
DFB      LOW{tmp2}
DFB      LOW{tmp1}
DFB      LOW{tmp0}
DFB      " us "
;
Var_OEIPD:
DFB      "Onde excitante "
;
DFB      " IPD "
DFB      LOW{oeipd2}
DFB      LOW{oeipd1}
DFB      LOW{oeipd0}
DFB      " us "
;
DFB      "Nouvelle valeur "
;
DFB      " "
DFB      LOW{tmp2}
DFB      LOW{tmp1}
DFB      LOW{tmp0}
DFB      " us "
;
Var_OIFRQ:
DFB      "Onde inhibitrice"
;
DFB      " FRQ "
DFB      LOW{oifrq3}
DFB      LOW{oifrq2}
DFB      LOW{oifrq1}
DFB      LOW{oifrq0}
DFB      " Hz "
;
DFB      "Nouvelle valeur "
;
DFB      " "
DFB      LOW{tmp3}
DFB      LOW{tmp2}
DFB      LOW{tmp1}
DFB      LOW{tmp0}
DFB      " Hz "
;
Var_OIAMP:
DFB      "Onde inhibitrice"
;
DFB      " AMP "
DFB      LOW{oiampi}
DFB      " "

```

```

        DFB      LOW{oiamp0}
        DFB      " mA "
;
        DFB      "Nouvelle valeur "
;
        DFB      "
        DFB      LOW{tmp1}
        DFB      "."
        DFB      LOW{tmp0}
        DFB      " mA "
;
Var_OIPWD:
        DFB      "Onde inhibitrice"
;
        DFB      " PWD "
        DFB      LOW{oipwd2}
        DFB      LOW{oipwd1}
        DFB      LOW{oipwd0}
        DFB      " us "
;
        DFB      "Nouvelle valeur "
;
        DFB      "
        DFB      LOW{tmp2}
        DFB      LOW{tmp1}
        DFB      LOW{tmp0}
        DFB      " us "
;
Var_OIIPD:
        DFB      "Onde inhibitrice"
;
        DFB      " IPD "
        DFB      LOW{oiipd2}
        DFB      LOW{oiipd1}
        DFB      LOW{oiipd0}
        DFB      " us "
;
        DFB      "Nouvelle valeur "
;
        DFB      "
        DFB      LOW{tmp2}
        DFB      LOW{tmp1}
        DFB      LOW{tmp0}
        DFB      " us "
;
Var_Mem:
        DFS      {64}          ; Pas de variables
;
Var_MemSauv:
        DFS      {64}          ; Pas de variables
;
Var_MemCharge_OE:
        DFB      LOW{memoire}
        DFB      ") FRQ "
        DFB      LOW{oeirq3t}
        DFB      LOW{oeirq2t}
        DFB      LOW{oeirq1t}
        DFB      LOW{oeirq0t}
        DFB      " Hz "
;
        DFB      "OE AMP "
        DFB      LOW{oeamp1t}
        DFB      "."

```

```

        DFB      LOW{oeamp0t}
        DFB      " mA "
;
        DFB      "* PWD "
        DFB      LOW{oepwd2t}
        DFB      LOW{oepwd1t}
        DFB      LOW{oepwd0t}
        DFB      " us "
;
        DFB      "** IPD "
        DFB      LOW{oeipd2t}
        DFB      LOW{oeipd1t}
        DFB      LOW{oeipd0t}
        DFB      " us "
;
Var_MemCharge_OI:
        DFB      LOW{memoire}
        DFB      ") FRQ "
        DFB      LOW{oifrq3t}
        DFB      LOW{oifrq2t}
        DFB      LOW{oifrq1t}
        DFB      LOW{oifrq0t}
        DFB      " Hz "
;
        DFB      "OI AMP "
        DFB      LOW{oiamp1t}
        DFB      "."
        DFB      LOW{oiamp0t}
        DFB      " mA "
;
        DFB      "* PWD "
        DFB      LOW{oipwd2t}
        DFB      LOW{oipwd1t}
        DFB      LOW{oipwd0t}
        DFB      " us "
;
        DFB      "** IPD "
        DFB      LOW{oiipd2t}
        DFB      LOW{oiipd1t}
        DFB      LOW{oiipd0t}
        DFB      " us "
;
Var_StimDuree:
        DFB      "Duree de      "
        DFB      "stimulation:  "
        DFB      " "
        DFB      LOW{tmp1}
        DFB      LOW{tmp0}
        DFB      " secondes  "
        DFB      " "
;
Var_StimActive:
        DFB      "* STIM. ACTIVE *"
        DFB      " "
        DFB      LOW{tmp1}
        DFB      LOW{tmp0}
        DFB      " secondes  "
        DFB      "[Une touche  "
        DFB      " pour arreter] "
;
; Fin de "LCD_TBL.ASM"
;-----

```