

Titre: Conception et évaluation d'un système de stimulation électrique
neurale dédié à la réhabilitation des fonctions vésicales

Auteur: Eric Schneider
Author:

Date: 2001

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Schneider, E. (2001). Conception et évaluation d'un système de stimulation
électrique neurale dédié à la réhabilitation des fonctions vésicales [Mémoire de
maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/7302/>

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

URL de PolyPublie: <https://publications.polymtl.ca/7302/>
PolyPublie URL:

**Directeurs de
recherche:** Mohamad Sawan
Advisors:

Programme: Non spécifié
Program:

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

UNIVERSITÉ DE MONTRÉAL

CONCEPTION ET ÉVALUATION
D'UN SYSTÈME DE STIMULATION ÉLECTRIQUE NEURALE
DÉDIÉ À LA RÉHABILITATION DES FONCTIONS VÉSICALES

ERIC SCHNEIDER

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLOME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)

AVRIL 2001



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: 0-612-90859-3

Our file Notre référence

ISBN: 0-612-90859-3

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.

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

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

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

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CONCEPTION ET ÉVALUATION
D'UN SYSTÈME DE STIMULATION ÉLECTRIQUE NEURALE
DÉDIÉ À LA RÉHABILITATION DES FONCTIONS VÉSICALES

Présenté par : SCHNEIDER Éric

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. SAVARIA Yvon, Ph.D., président

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

M. YAHIA L'Hocine, Ph.D., membre

DÉDICACE

*À mes parents Claude et Dominique,
ma sœur Pascale et mon petit frère Brice.*

REMERCIEMENTS

Je tiens avant tout à remercier mon directeur de recherche, Mohamad Sawan, qui a bien voulu m'accepter au sein de son équipe de recherche et qui a su me conseiller et me guider tout au long de ma maîtrise.

Un autre remerciement particulier à Gaétan Décarie, technicien du département de génie électrique de l'École Polytechnique de Montréal, pour son grand support technique et sans qui les prototypes n'auraient sûrement pas été réalisés avec autant de professionnalisme et de minutie.

Tous mes remerciements,

À l'équipe médicale du département d'urologie de l'Université McGill et aux techniciens du centre des ressources animales de l'Université McGill qui ont contribué au bon déroulement de nos études en phase expérimentale.

Aux membres du jury qui ont bien voulu prendre le temps de s'intéresser au sujet du présent mémoire et ont contribué à en améliorer la qualité.

À tous les techniciens et techniciennes du département de génie électrique de l'École Polytechnique qui m'ont été d'un grand support tout au long de mes travaux.

À mes collègues et amis de l'équipe PolySTIM qui ont su créer une ambiance de travail agréable et ont contribué à mon intégration dans ce nouveau pays.

Enfin, je remercie la Fondation Canadienne des Maladies du Rein (FCMR) et le Groupe de Recherche Inter-universitaire en Architecture des Ordinateurs (GRIAO) pour leurs supports financiers tout au long de mes études.

RÉSUMÉ

Les travaux présentés dans ce document trouvent leur application dans le traitement d'organes paralysés chez des patients devenus paraplégiques suite à une lésion de la moelle épinière. Plus particulièrement, nous abordons le thème des dysfonctions vésicales, où le patient n'est plus en mesure de provoquer une miction volontaire lorsque sa vessie est pleine et souffre de complications vésicales liées à une hyperactivité réflexe du muscle de la vessie (hyperréflexie).

Nous proposons ici un dispositif électronique implantable pour la réhabilitation des fonctions vésicales par la stimulation électrique neurale. Rétablir les fonctions vésicales signifie permettre de nouveau une miction volontaire, réduire voire éliminer les problèmes d'hyperréflexie et idéalement pouvoir faire un suivi in vivo du niveau de remplissage de la vessie ainsi que de l'état des tissus stimulés. Le système est constitué d'un microstimulateur dédié à être implanté à l'intérieur du corps sous la peau du patient et d'un contrôleur externe permettant de communiquer avec le stimulateur sous-cutané et de l'alimenter en énergie via un lien inductif opérant à haute fréquence.

Deux prototypes de microstimulateur, supportant chacun deux des trois fonctionnalités souhaitées, ont été réalisés sur des circuits imprimés en se servant de composants électroniques miniatures disponibles commercialement. Chacun de ces prototypes étant de taille suffisamment petite pour faire l'objet de tests in vivo chez l'animal.

Le premier prototype regroupe une fonction de stimulation dite sélective permettant la miction volontaire et une fonction de mesure de l'impédance du contact entre le nerf et les électrodes. Des tests en laboratoire ont démontré la fonctionnalité souhaitée du système.

Le second prototype dispose lui aussi de la fonction de stimulation sélective en plus d'une fonction de stimulation dite permanente qui devrait permettre de réduire les problèmes d'hyperréflexie et tous les effets qui y sont rattachés. La stimulation permanente est une stimulation à très basse fréquence et très basse amplitude qui nécessite une alimentation par pile. Par contre la stimulation sélective est une stimulation bi-fréquence, ponctuelle, précise et à plus forte amplitude, qui doit être parfaitement contrôlée de l'extérieur.

Huit exemplaires du second prototype ont fait l'objet d'une étude expérimentale à long terme sur des animaux de laboratoire (chiens), afin de caractériser la fiabilité et la fonctionnalité de l'implant. Le suivi médical a été assuré par des médecins spécialistes en urologie et des techniciens du centre des ressources animales de l'Université McGill. Les résultats préliminaires de cette étude montrent que le microstimulateur proposé et la technique chirurgicale exercée répondent aux attentes de notre équipe (PolySTIM) pour la réduction des problèmes d'hyperréflexie chez les paraplégiques.

Les résultats obtenus à partir des tests en laboratoire et des expérimentations chez l'animal démontrent que les méthodes de stimulation sélective, de stimulation permanente et de surveillance in vivo présentent un grand intérêt sur le plan médical. De plus, l'utilisation de ces microstimulateurs ne se limiterait pas à la réhabilitation vésicale,

ces dispositifs pourraient tout aussi bien être utilisés dans d'autres situations de paralysie qui nécessitent les deux stimulations neurales sélective et permanente. Toutefois, quelques parties du système de stimulation proposé exigent des améliorations avant qu'il ne soit utilisable pour des applications humaines, à titre d'exemples, citons le procédé d'isolation électrique des implants face au milieu biologique corrosif dans lequel ils sont utilisés, la robustesse des contrôleurs externes et la fiabilité des circuits critiques du microstimulateur.

ABSTRACT

The aim of this master thesis is to study the rehabilitation of paralyzed organs in paraplegic patients that have undergone spinal cord injuries. Among the affected organs, the urinary bladder, where the patient is unable to voluntarily evacuate the urine from his bladder when it is full, and he often suffers from many vesical complications related to the bladder muscle hyperactivity (hyperreflexia).

An implantable electronic device, dedicated to recover the bladder functions using neural electrical stimulation, is proposed. Restoring urinary functions allows voluntary bladder voiding, reducing or even suppressing hyperreflexia, and monitoring bladder filling as well as the status of stimulated tissues. The stimulation system is composed of a subcutaneous implanted microstimulator, and an external controller providing digital information, supplying clock and energy to the implant. Power is transferred through the skin using a high frequency inductive coupling link.

Two versions of microstimulators have been implemented, each one enabling 2 out of the 3 curing functionalities necessary to a complete bladder rehabilitation. Those implants have been implemented on custom printed circuit boards using surface mounted commercially available electronic devices, allowing to assemble implants small enough to be implanted in animals and validate both the electronic and the surgical issues.

The first version includes the selective stimulation function, which enables voluntary voiding of the bladder and in vivo monitoring functions that allow to measure

the impedance of the electrodes-nerve interface. Laboratory tests of the implant show a complete functionality and flexibility of the device.

The second version includes the same selective stimulation for bladder voiding, in addition to a new permanent stimulation technique to reduce (or cancel) the bladder hyperreflexia, curing other related diseases. Permanent stimulation is a low frequency, low amplitude and all day long stimulation that needs to be battery powered. On the other hand, selective stimulation is a bi-frequency, punctual, precise and at a higher amplitude stimulation that is well controlled and powered from the outside.

Eight prototypes of the second version have been used in an experimental evaluation in animals (dogs) to characterize the reliability and functionality of the new implant in a real application environment. Medical assistance was conducted by urologists and technicians of the animal resources center of McGill University. Preliminary results of the study show that the proposed device and the surgical process provide significant improvement for bladder hyperreflexia curing.

The experimental validation on animals confirmed that selective stimulation, permanent stimulation and in vivo monitoring have great medical promises for bladder rehabilitation. Furthermore, such microstimulator could also be used for any other organ dysfunction that needs both neural selective and permanent electrical stimulation. However, insulation process of the battery against corrosive biological environment need to be optimized, the strength of the external controller must be increased and critical circuits of the microstimulator have to be secured to ensure better reliability.

TABLE DES MATIÈRES

DÉDICACE.....	iv
REMERCIEMENTS.....	v
RÉSUMÉ.....	vi
ABSTRACT	ix
TABLE DES MATIÈRES	xi
LISTE DES FIGURES	xiv
LISTE DES TABLEAUX.....	xix
LISTE DES ANNEXES	xx
LISTE DES SIGLES ET ABRÉVIATIONS.....	xxi
INTRODUCTION.....	1
CHAPITRE 1 NEUROSTIMULATEUR À LIEN BIDIRECTIONNEL	8
1.1 Introduction	8
1.2 Spécifications et contraintes.....	9
1.3 Principe d'évaluation d'impédance.....	11
1.4 Design du NLB.....	12
1.4.1 Protocole de communication.....	13
1.4.2 Architecture détaillée	14
1.4.3 Étage de stimulation et de mesure.....	16
1.4.4 Lien RF bidirectionnel	18
1.4.5 Contrôleur intégré sur FPGA	21

1.5	Réalisation du prototype.....	25
1.6	Conclusion.....	26
CHAPITRE 2 NEUROSTIMULATEUR SÉLECTIF ET PERMANENT.....		27
2.1	Introduction	27
2.2	Spécifications	27
2.3	Architectures envisagées	29
2.4	Design du NSP	31
2.4.1	Modes de fonctionnement	32
2.4.2	Protocole de communication.....	34
2.4.3	Architecture détaillée	36
2.4.4	Étage de réception et étage de stimulation.....	38
2.4.5	Contrôleur intégré sur FPGA	39
2.4.6	Contrôleur intégré sur PIC	43
2.5	Réalisation du prototype.....	46
2.6	Conclusion.....	48
CHAPITRE 3 CONTRÔLEURS EXTERNES		50
3.1	Introduction	50
3.2	Couplage magnétique et transfert de puissance	51
3.3	Contrôleurs de test flexibles.....	55
3.4	Contrôleurs portatifs.....	57
3.5	Module d'affichage d'impédance.....	59
3.6	Stabilisation de la porteuse par contrôle numérique	61
3.7	Conclusion.....	64
CHAPITRE 4 EXPÉRIMENTATIONS ET RÉSULTATS		66
4.1	Introduction	66
4.2	Tests fonctionnels et analyse des performances du NLB.....	67
4.2.1	Caractérisation de la stimulation sélective	68
4.2.2	Caractérisation de la mesure d'impédance.....	70

4.3	Tests fonctionnels et analyse des performances du NSP	74
4.3.1	Caractérisation de la stimulation sélective	76
4.3.2	Caractérisation de la stimulation permanente	78
4.4	Préparation du matériel pour les tests en phase chronique.....	80
4.4.1	Assemblage des microstimulateurs	80
4.4.2	Les électrodes de stimulation	82
4.5	Protocole expérimentale des tests en phase chronique.....	83
4.5.1	Protocole chirurgical et implantation.	84
4.5.2	Suivi médical en phase chronique.....	85
4.6	Résultats expérimentaux et analyse de la fiabilité	88
4.7	Conclusions.....	94
CONCLUSION.....		96
BIBLIOGRAPHIE		100
ANNEXES.....		108

LISTE DES FIGURES

Figure 1.1 :	Forme d'onde d'un signal de stimulation sélective	10
Figure 1.2 :	Formes d'ondes des signaux pour la mesure d'impédance	13
Figure 1.3 :	Structures des trames de données du NLB.....	14
Figure 1.4 :	Schéma bloc du NLB	15
Figure 1.5 :	Schéma électrique du VCO et de la source de courant	17
Figure 1.6 :	Schéma électrique de l'oscillateur FSK et de l'adaptateur d'impédance.....	20
Figure 1.7 :	Schéma du décodeur Manchester.....	21
Figure 1.8 :	Schéma bloc du contrôleur sur FPGA du NLB.....	22
Figure 1.9 :	Photographie du stimulateur NLB à l'échelle 1/1	26
Figure 2.1 :	Forme d'onde d'un signal de stimulation permanente	28
Figure 2.2 :	Propositions d'architectures pour le NSP	31
Figure 2.3 :	Schéma bloc simplifié du système de stimulation NSP	32
Figure 2.4 :	Structures des trames de données du NSP	35
Figure 2.5 :	Schéma bloc détaillé du NSP	36
Figure 2.6 :	Diagramme d'états simplifié du NSP	37
Figure 2.7 :	Schéma bloc de la file à détection d'entête du FPGA du NSP	39
Figure 2.8 :	Diagramme d'états de la file à détection d'entête du FPGA du NSP.....	40
Figure 2.9 :	Schéma bloc du contrôleur intégré sur FPGA pour le NSP	42
Figure 2.10 :	Diagramme d'états du contrôleur intégré sur PIC pour le NSP.....	43
Figure 2.11 :	Schéma de l'interrupteur d'alimentation du PIC pour le NSP	45
Figure 2.12 :	Photographie du NSP à l'échelle 1/1.....	48
Figure 3.1 :	Schéma bloc générique d'un contrôleur externe.....	50
Figure 3.2 :	Topologie d'un circuit à couplage inductif.....	52
Figure 3.3 :	Variation du rapport V_{out}/V_{in} pour différentes valeurs de k	52
Figure 3.4 :	Topologie d'un amplificateur de classe D	54

Figure 3.5 :	Schéma électrique de l'émetteur ASK des contrôleurs externes	55
Figure 3.6 :	Schéma bloc de l'interface PC-modulateur du CTF	56
Figure 3.7 :	Exemple d'écran de l'interface utilisateur du CTF	57
Figure 3.8 :	Schémas blocs des contrôleurs portatifs simplifié (a) et flexible (b)	58
Figure 3.9 :	Schéma bloc du module de réception de données FSK	60
Figure 3.10 :	Schéma bloc détaillé d'un CTF à contrôle numérique de la porteuse	63
Figure 4.1 :	Temps de montée de l'alimentation (Ch1) et signal de stimulation sélective (Ref1) pour le NLB	69
Figure 4.2 :	Courbe de réponse réelle (a) et modélisée (b) du VCO	70
Figure 4.3 :	Exemple de signaux intervenants dans la mesure d'impédance du NLB : oscillation du VCO (Ch2) et signal de stimulation (Ch1).....	74
Figure 4.4 :	Temps de montée de l'alimentation (Ch1) et signal de stimulation sélective (Ref1) pour le NSP	77
Figure 4.5 :	Signal de stimulation permanente (Ref1) et horloge du PIC(Ch2) pour le NSP	79
Figure 4.6 :	Photographie du NSP prêt à être implanté	82
Figure 4.7 :	Photographie d'une électrode à sutures (a) et d'une électrode à alliage super-élastique (b)	83
Figure 4.8 :	Identification des nerfs de la racine sacré	85
Figure 4.9 :	Pression dans la vessie (Pves) et dans l'urètre (Pura) lors d'une stimulation sélective avec le NSP	87
Figure 4.10 :	Pression dans la vessie (Pves) et dans l'urètre (Pura) lors d'une stimulation permanente par le NSP	87
Figure 4.11 :	Bilan de fiabilité des NSP pendant l'année d'évaluation.....	89
Figure 4.12 :	Cheminement des travaux expérimentaux et principaux événements survenus.....	92
Figure A.1 :	Schéma électrique du NLB	109
Figure A.2 :	Dessin des masques à l'échelle 15/10 du NLB	110
Figure A.3 :	Répartition des broches pour le FPGA du NLB.....	111

Figure A.4 :	Schéma logique de niveau hiérarchique supérieur pour le FPGA du NLB.....	112
Figure A.5 :	Schéma logique de la file à détection d'entête pour le FPGA du NLB.....	113
Figure A.6 :	Schéma logique d'un registre à décalage pour le FPGA du NLB.....	114
Figure A.7 :	Schéma logique du générateur d'impulsions haute fréquence pour le FPGA du NLB.....	114
Figure A.8 :	Schéma logique du générateur d'impulsions basse fréquence pour le FPGA du NLB.....	115
Figure A.9 :	Schéma logique du contrôleur de largeur d'impulsions basse fréquence pour le FPGA du NLB.....	115
Figure A.10 :	Schéma logique du contrôleur de largeur d'impulsions haute fréquence pour le FPGA du NLB.....	116
Figure A.11 :	Schéma logique du sélecteur d'impulsions haut ou basse fréquence pour le FPGA du NLB	116
Figure A.12 :	Schéma logique du multiplexeur de sélection de l'amplitude de stimulation pour le FPGA du NLB	117
Figure A.13 :	Schéma logique du décodeur Manchester pour le FPGA du NLB	118
Figure A.14 :	Schéma logique du compteur de fréquence pour le FPGA du NLB	118
Figure A.15 :	Schéma logique du convertisseur parallèle à série asynchrone pour le FPGA du NLB.....	119
Figure B.1 :	Schéma électrique initial du NSP.....	125
Figure B.2 :	Schéma électrique corrigé du NSP.....	126
Figure B.3 :	Dessin des masques initial à l'échelle 2/1 du NSP.....	127
Figure B.4 :	Modifications apportées au dessin des masques du NSP.....	128
Figure B.5 :	Répartition des broches pour le FPGA du NSP	130
Figure B.6 :	Schéma logique de niveau hiérarchique supérieur pour le FPGA du NSP.....	131

Figure B.7 :	Schéma logique de la file à détection d'entête pour le FPGA du NSP.....	132
Figure B.8 :	Schéma logique du registre à décalage des paramètres de stimulation pour le FPGA du NSP	133
Figure B.9 :	Schéma logique d'un registre à décalage pour le FPGA du NSP	134
Figure B.10 :	Schéma logique du calculateur de CRC pour le FPGA du NSP	135
Figure B.11 :	Schéma logique du détecteur de mode pour le FPGA du NSP	136
Figure B.12 :	Schéma logique du détecteur de cinq et de six "1" consécutifs dans une trame pour le FPGA du NSP	137
Figure B.13 :	Schéma logique du détecteur de dépassement de longueur de trame pour le FPGA du NSP	137
Figure B.14 :	Schéma logique du décodeur Manchester pour le FPGA du NSP	138
Figure B.15 :	Schéma logique du temporisateur de synchronisation pour le FPGA du NSP	138
Figure B.16 :	Schéma logique du générateur d'impulsions haute fréquence pour le FPGA du NSP	139
Figure B.17 :	Schéma logique du générateur d'impulsions basse fréquence pour le FPGA du NSP	139
Figure B.18 :	Schéma logique du contrôleur de largeur d'impulsions pour le FPGA du NSP	140
Figure B.19 :	Schéma logique du contrôleur de commutation du sens du courant de stimulation pour le FPGA du NSP	140
Figure B.20 :	Schéma logique du générateur de commandes pour DAC sériel pour le FPGA du NSP	141
Figure B.21 :	Schéma logique du contrôleur des Entrées/Sorties pour le FPGA du NSP.....	141
Figure C.1 :	Schéma électrique de l'émetteur RF utilisé pour les contrôleurs externes.....	161
Figure C.2 :	Schéma électrique de l'interface parallèle du CTF	162

Figure C.3 :	Répartition des broches du FPGA de l'interface parallèle du CTF.....	163
Figure C.4 :	Schéma logique de niveau hiérarchique supérieur du PFPGA de l'interface parallèle du CTF	164
Figure C.5 :	Schéma électrique du CPS	187
Figure D.1 :	Schéma électrique du module d'affichage d'impédance.....	205
Figure D.2 :	Schéma électrique du module de stabilisation de la porteuse	206
Figure D.3 :	Répartition des broches du FPGA du module de stabilisation de la porteuse	207
Figure D.4 :	Schéma logique de niveau hiérarchique supérieur du FPGA du module de stabilisation de la porteuse.....	208
Figure D.5 :	Schéma logique du diviseur de fréquence par 20 du FPGA du module de stabilisation de la porteuse.....	209
Figure D.6 :	Schéma logique du diviseur de fréquence programmable du FPGA du module de stabilisation de la porteuse.....	209
Figure D.7 :	Schéma logique du compteur programmable synchrone du FPGA du module de stabilisation de la porteuse.....	210
Figure D.8 :	Schéma logique du contrôleur de registre à décalage de 12 bits du FPGA du module de stabilisation de la porteuse	211
Figure D.9 :	Schéma logique d'un registre 1 bit à double latch du FPGA du module de stabilisation de la porteuse.....	211
Figure D.10 :	Schéma logique du registre à décalage de 12 bits du FPGA du module de stabilisation de la porteuse.....	212
Figure D.11 :	Schéma logique de l'accumulateur 96 bits de données sérielles du FPGA du module de stabilisation de la porteuse	213
Figure D.12 :	Schéma logique du contrôleur de registre à décalage de 96 bits du FPGA du module de stabilisation de la porteuse	214
Figure D.13 :	Schéma logique du codeur Manchester du FPGA du module de stabilisation de la porteuse	214

LISTE DES TABLEAUX

Tableau 1.1 : Description des paramètres pour la stimulation sélective	10
Tableau 1.2 : Les paramètres d'une trame et leur expression littérale	23
Tableau 2.1 : Description des paramètres pour la stimulation permanente	29
Tableau 2.2 : Codes et conventions du protocole de communication avec le NSP	35
Tableau 2.3 : Les principaux registres du PIC du NSP.....	44
Tableau 4.1 : Comparaison des performances du MNS et du NLB pour la stimulation sélective.....	68
Tableau 4.2 : Autres caractéristiques importantes du NLB	69
Tableau 4.3 : Coefficients de proportionnalité du VCO en fonction des intervalles de fréquences	71
Tableau 4.4 : Tableau comparatif des performances du MNS et du NSP pour la stimulation sélective.....	76
Tableau 4.5 : Autres caractéristiques importantes du NSP	77
Tableau 4.6 : Performances du NSP lors de la stimulation permanente	78
Tableau 4.7 : Autres informations importantes sur la fiabilité des NSP	90
Tableau A.1 : Liste des circuits intégrés et des semiconducteurs du NLB.....	110
Tableau B.1 : Liste des circuits intégrés et des semiconducteurs du NSP.....	129

LISTE DES ANNEXES

ANNEXE A	Schémas et codes du NLB.....	108
ANNEXE B	Schémas et codes du NSP.....	124
ANNEXE C	Schémas et codes des contrôleurs externes	160
ANNEXE D	Schémas du module d'affichage de l'impédance et du module de stabilisation de la porteuse	204

LISTE DES SIGLES ET ABRÉVIATIONS

AM	Amplitude modulation
Amp	Amplitude
AOP	Amplificateur Opérationnel
ASK	Amplitude shift keying
CAN	Convertisseur analogique à numérique
CNA	Convertisseur numérique à analogique
CPF	Contrôleur portative flexible
CPS	Contrôleur portative simplifié
CRC	Code redundancy check
CTF	Contrôleur de test flexible
FPGA	Field programmable gate array
Freq	Frequency
FSK	Frequency shift keying
HFA	High frequency amplitude
HFP	High frequency period
HFW	High frequency width
LCD	Liquid crystal display
LFA	Low frequency amplitude
LFP	Low frequency period
LFW	Low frequency width
MNS	Microstimulateur neural sélectif
NLB	Neurostimulateur à lien bidirectionnel
NSP	Neurostimulateur sélectif et permanent
PC	Personal computer
PCB	Printed circuit board
PIC	Programmable integrated controller

PLL	Phase lock loop
PW	Pulse width
RF	Radio frequency
UART	Universal asynchronous receiver transmitter
VCO	Voltage controlled oscillator

INTRODUCTION

Il n'est plus surprenant de nos jours de voir une personne vivre normalement avec un dispositif électronique implanté assistant certaines de ses fonctions vitales. On pense notamment au stimulateur cardiaque (pacemaker), à l'implant cochléaire, aux stimulateurs musculaires pour les membres paralysés, au stimulateur urinaire et on envisage maintenant de réaliser des stimulateurs corticaux pour le rétablissement partiel de la vision [11], [12], [45], [48].

Les troubles du système urinaire chez les patients paraplégiques ont fait l'objet d'études avancées ces dernières années [7], [21]. Le cathétérisme intermittent et le sondage permanent permettaient de contourner le problème, mais les irritations et les infections liées à cette technique et l'avènement de l'électronique ont encouragé divers équipes de recherche à analyser le sujet en profondeur [9], [24], [68]. Le monde de la recherche en est arrivé à une compréhension très précise du fonctionnement du système urinaire et a développé de nombreuses techniques souvent basées sur la stimulation électrique permettant un rétablissement fonctionnel acceptable [16], [31], [59].

Les dysfonctions vésicales chez les paraplégiques sont directement liées à l'interruption de l'innervation sensitive et motrice passant par la moelle épinière, unique lien de communication entre le cerveau et les membres inférieurs. Un sujet normal éprouve un besoin d'uriner lorsque sa vessie est pleine, le cerveau commande alors volontairement la contraction du muscle de la vessie (le detrusor) et la relaxation du sphincter permettant ainsi la miction. En plus du contrôle volontaire de la miction, le

cerveau assure aussi une inhibition permanente des contractions réflexes de la vessie qui peuvent survenir lors de son remplissage [37], [60].

La majorité des travaux de recherche effectués sur le système urinaire, se sont principalement concentrés sur le contrôle de la miction volontaire. Les expériences de stimulations électriques ont naturellement commencées par une stimulation directe du detrusor [29], [63]. Cette technique nécessitait beaucoup d'énergie, un grand nombre d'électrodes minutieusement placées et impliquait des contraintes mécaniques trop dommageables pour les électrodes [28], [69], [73]. Elle fut très vite abandonnée pour la stimulation électrique neurale qui permettait de réutiliser la structure de l'innervation naturelle de la vessie. Différents sites de stimulation sont alors possibles : les nerfs pelviens, terminaisons nerveuses innervant directement le detrusor; la racine sacrée (S1 à S5), extrémité de la moelle épinière au niveau du bassin innervant le detrusor et le sphincter; la moelle épinière, regroupant l'ensemble des influx nerveux du corps.

L'application de la stimulation électrique par le biais d'électrodes sur les nerfs pelviens est rendue difficile par l'arborescence complexe du plexus pelviens autour de la vessie. De plus, ces nerfs ne supportent pas des stimulations électriques de longue durée, car ils risquent facilement d'être endommagés de façon irréversible [33], [34], [69]. La stimulation de la moelle épinière nécessite des électrodes très précises pour atteindre le centre d'activation de la vessie et ne présente pas une solution viable à long terme, à cause du déplacement des électrodes [36], [38], [75]. Reste finalement la stimulation des nerfs de la racine sacrée qui semble être la solution la plus prometteuse [5], [26], [69].

Le nerf sacré S2 a été identifié comme le principal responsable dans le contrôle de la vessie. Ce dernier est constitué de deux types de fibres : les fibres parasympathiques, petites fibres innervant principalement le detrusor et les fibres somatiques, plus grosses, innervant principalement le sphincter [10], [57]. Une stimulation électrique conventionnelle du nerf sacré S2 active la conduction des 2 types de fibres et provoque une contraction simultanée de la vessie et du sphincter. Ce phénomène appelé dyssynergie empêche la miction et augmente la pression vésicale qui est alors très dommageable pour les reins [41], [74], [76]. Une première solution implique une intervention chirurgicale, appelée rhizotomie, qui consiste à sectionner les terminaisons nerveuses d'un ou plusieurs nerfs sacrés qui innervent le sphincter [11], [55], [61], [72]. Cependant, cette intervention est irréversible et a des effets secondaires sur d'autres fonctions vitales liées au système urinaire [43], [44]. Il est donc préférable d'utiliser une stimulation sélective qui permet de stimuler séparément les 2 types de fibres en bloquant la conduction des grosses fibres tout en maintenant celle des petites fibres.

Trois techniques pour le blocage de la conduction des grosses fibres du nerf sacré S2 sont actuellement examinées : le blocage anodique, le blocage par collision et le blocage par stimulation à haute fréquence [6], [12], [25], [50]. Cette dernière méthode est celle qui a été retenue pour notre présente étude et a été proposée à l'origine par notre équipe de recherche [53], [65]. Cette méthode consiste à utiliser deux stimuli à haute et basse fréquences qui sont appliqués à l'aide d'une seule électrode bipolaire. L'objectif est d'inhiber la contraction du sphincter en bloquant la conduction des grosses fibres par un

signal de haute fréquence (600Hz) et d'activer la contraction du detrusor par un signal de basse fréquence (30Hz).

La récupération du contrôle volontaire de la miction ne règle pas les autres traumatismes vésicaux qui résultent de l'isolement de la vessie suite à une lésion de la moelle épinière. En effet, lorsque le lien entre le cerveau et la vessie est interrompu, les contractions réflexes ne peuvent plus être inhibées et ont tendance à s'amplifier. Cette hyperactivité, aussi appelée hyperréflexie, provoque des problèmes d'incontinence entre les périodes de miction, mais implique surtout des suppressions répétitives dans le canal rénal qui entraînent généralement de graves infections et des dommages irréversibles du système vésical [74], [76]. Le traitement direct de l'incontinence par la stimulation électrique chez des patients non paraplégiques a déjà fait l'objet de nombreuses études dont un système déjà commercialisé [45]. Dans notre cas, il ne s'agit pas de résoudre directement l'incontinence, mais plutôt de traiter l'hyperréflexie du detrusor, qui est la cause de cette incontinence et de bien d'autres problèmes chez les paraplégiques.

Le mécanisme de régulation des réflexes de la vessie par le cerveau n'a à ce jour pas encore été clairement identifié, mais la racine sacrée et certaines substances chimiques comme les neuropeptides semblent jouer un rôle essentiel dans cette régulation [22], [64], [62]. Tout porte à croire qu'une stimulation électrique de basse fréquence et de faible amplitude au niveau de la racine sacrée permettrait de contrôler la concentration de ces neuropeptides et ainsi de réduire l'hyperréflexie. Les paramètres exacts de cette stimulation électrique ne sont toutefois pas encore clairement identifiés. Comment alors déterminer ces paramètres autrement que par une étude expérimentale à long terme. Il est

donc primordial de disposer d'un système électronique programmable et implantable permettant à la fois d'effectuer une excitation permanente pour le traitement de l'hyperréflexie et une stimulation sélective pour le contrôle de la miction. Nous nous retrouverions alors avec un dispositif en mesure d'effectuer des commandes semblables à celles générées par le cerveau chez les sujets ayant des fonctions urinaires normales.

Les expériences en phase aiguë (le temps d'une opération) exigent une attention particulière, mais elles ne nécessitent pas une surveillance accrue de l'état des tissus stimulés et des électrodes utilisées. Pour des expériences de plus longue durée, dite en phase chronique, il n'est pas rare de constater une baisse inexplicée de l'efficacité de stimulation ou encore une interruption soudaine du fonctionnement de l'implant. Ces difficultés peuvent être liées à des bris mécaniques, à la fatigue du nerf stimulé ou encore à des changements dans les propriétés électriques du contact nerf-électrodes, qui rendent la stimulation temporairement inefficace [3], [19]. Dans certains cas, une intervention chirurgicale est requise, mais dans la majorité des cas, une période de repos ou un traitement médicamenteux suffit. Il a été jugé nécessaire, d'un point de vue médical et d'un point de vue optimisation de la stimulation appliquée, de pouvoir surveiller l'évolution de l'implant et de certains paramètres caractérisant les interfaces électrodes-tissus pendant toute la durée d'une étude en phase chronique.

Un microstimulateur neural sélectif (MNS) à blocage par haute fréquence a déjà été développé et validé par notre équipe de recherche [8], [65]. L'efficacité et la fiabilité du système électronique réalisé avec des circuits discrets et programmable, ne nécessitait pas de modification majeure. Toutefois les tests en phase chronique ont révélé

l'importance de concevoir un nouveau système implantable capable de mesurer l'impédance de l'interface nerf-électrodes in-vivo et de retourner la valeur mesurée vers un afficheur externe. Le nouveau neurostimulateur à lien bidirectionnel (NLB) empruntera son architecture à son prédécesseur (MNS) pour des raisons évidente de délais et de simplicité.

Parallèlement, une forte demande du milieu de la recherche médicale, nous a motivé à développer une troisième génération du MNS. Cette dernière inclut une fonctionnalité de stimulation permanente pour une réhabilitation complète du système urinaire (miction et hyperréflexie). Ce nouveau neurostimulateur sélectif et permanent (NSP) assurant les deux types de stimulation, sélective et permanente, devra autant que possible s'appuyer sur l'architecture des deux premières générations, tout en étant aussi configurable pour l'une ou l'autre des fonctions.

Nous présentons au Chapitre 1 le détail des nouvelles fonctionnalités intégrées au neurostimulateur à lien bidirectionnel (NLB) et les contraintes de conception liées à son entière compatibilité avec la première génération de microstimulateur neural sélectif (MNS). L'architecture interne ainsi que le principe de mesure d'impédance sont expliqués et le prototype implantable du NLB réalisé sur circuit imprimé est finalement présenté. Le Chapitre 2 introduit de la même façon les nouvelles caractéristiques du neurostimulateur sélectif et permanent (NSP) et clarifie la notion de stimulation permanente par rapport à la stimulation sélective. Les contraintes de gestion d'énergie introduites par la double source d'alimentation en énergie du système et leurs conséquences sur la nouvelle architecture du microstimulateur sont largement expliquées

et analysées. Le prototype implantable du NSP réalisé sur circuit imprimé est finalement présenté. Les différents types de contrôleurs externes essentiels au fonctionnement du système de stimulation complet sont exposés au Chapitre 3. On y présente aussi le module d'affichage externe utilisé pour la lecture de l'impédance avec le NLB, ainsi qu'un module améliorant la stabilité de la fréquence de la porteuse du lien inductif utilisé pour l'alimentation en énergie des microstimulateurs implantés. Les résultats de la caractérisation des performances de chacun des nouveaux microstimulateurs ainsi que les résultats d'une étude expérimentale en phase chronique chez l'animal de 9 exemplaires du NSP sont exposés au Chapitre 4. Finalement, nous dressons dans une conclusion un bilan sur les systèmes réalisés et les expériences acquises au cours de ces travaux, et nous terminons par une série de recommandations aussi bien sur la conception des futurs systèmes de stimulation que sur la méthodologie à suivre pour des futures expérimentations sur les dysfonctions du système urinaire.

CHAPITRE 1

NEUROSTIMULATEUR À LIEN BIDIRECTIONNEL

1.1 Introduction

Notre équipe de recherche, travaille depuis plusieurs années sur l'étude des effets à long terme de la stimulation électrique neurale de la vessie. Le dernier système de stimulation implantable, appelé microstimulateur neural sélectif, que nous avons développé (MNS), utilise la technique de stimulation sélective par blocage haute fréquence [53], [65]. Les tests en phase chronique effectués sur une dizaine d'animaux (chiens) en collaboration avec le département d'urologie de l'Université McGill ont confirmé l'efficacité de cette méthode de stimulation [8]. Le volume moyen d'urine évacué sur l'ensemble des sujets traités était supérieur à 90% du volume totale d'urine contenue dans la vessie, ce qui représente une amélioration de 50% par rapport aux autres types de stimulations électriques conventionnelles. Ces tests ont cependant fait apparaître des variations dans la sensibilité du nerf aux stimuli. Vraisemblablement, nous pouvons attribuer cette sensibilité à des variations de l'impédance de l'interface nerf-électrodes. Il est alors nécessaire de modifier les paramètres de stimulation pour s'assurer de toujours injecter la même quantité de charge dans le nerf. De plus, il arrive que les contacts des électrodes se brisent après quelques mois d'implantation et que du point de vue externe, il ne soit pas possible de déterminer s'il s'agit d'un épuisement (fatigue) du nerf, d'une défaillance du microstimulateur ou encore d'un bris d'électrodes. De telles constatations

nous ont convaincu de mettre au point une nouvelle version de ce microstimulateur sélectif en y incluant un dispositif en mesure d'évaluer in-vivo la composante résistive de l'interface nerf-électrodes. Ce chapitre présente en détail les principes de mesure, de commande et de transmission de ce neurostimulateur à lien bidirectionnel (NLB).

1.2 Spécifications et contraintes

La spécification principale de ce nouveau microstimulateur est de réutiliser intégralement la structure déjà existante. D'une part, parce qu'elle répondait bien aux besoins d'expérimentations et d'autre part parce qu'on souhaitait réduire le temps de développement. Nous présentons très succinctement ci-dessous les principales caractéristiques du premier microstimulateur neural sélectif (MNS) réalisé dans notre équipe.

Le MNS a été réalisé sur un circuit imprimé avec des composants discrets et un circuit intégré programmable. La partie centrale de l'implant est un circuit numérique implémenté dans un FPGA (Field Programmable Gate Array), il contrôle un seul canal de sortie analogique servant à générer des stimuli de courant bipolaires à deux fréquences indépendantes.

La Figure 1.1 représente une forme d'onde typique d'un signal de stimulation sélective généré par le MNS. Les six paramètres identifiés sont programmables à distance par un contrôleur externe. Ces paramètres sont :

- LFP (low frequency period) : période séparant 2 stimuli à basse fréquence, aussi référé, par abus de langage, comme la fréquence des stimuli basse fréquence

- HFP (high frequency period) : période séparant 2 stimuli à haute fréquence, aussi référé, par abus de langage, comme la fréquence des stimuli haute fréquence
- LFW (low frequency width) : largeur des stimuli basse fréquence
- HFW (high frequency width) : largeur des stimuli haute fréquence
- LFA (low frequency amplitude) : amplitude des stimuli basse fréquence
- HFA (high frequency amplitude) : amplitude des stimuli haute fréquence

Le Tableau 1.1 contient la description de ces paramètres incluant leur plage utile ainsi que la résolution autour de leur valeur typique.

Tableau 1.1 : Description des paramètres pour la stimulation sélective

Paramètre	Basse fréquence			Haute fréquence		
	Amplitude	Fréquence (1/Période) ²	Largeur d'impulsion	Amplitude	Fréquence (1/Période) ²	Largeur d'impulsion
Acronyme	LFA	LFP	LFW	HFA	HFP	HFW
Unité	mA	Hz	μs	mA	Hz	μs
Plage	0 – 2	18 – 2000	3.3 - 210	0 - 2	290 - 2000	3.3 - 210
Résolution ¹	8 μA	3 mHz	3.3 μs	8 μA	1.2 Hz	3.3 μs
Valeur typique	0.9	30	175	1.3	600	100

¹ Résolution autour d'une valeur typique (0.9mA, 30Hz, 175μs, 1.3mA, 600Hz, 100μs)
² Bien que les paramètres LFP et HFP correspondent aux périodes des stimuli, il est plus commode de considérer la fréquence (inverse de la période) des stimuli pour définir la forme d'onde

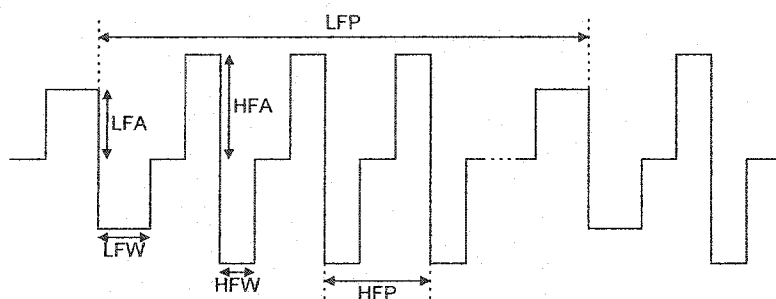


Figure 1.1 : Forme d'onde d'un signal de stimulation sélective

Le MNS sous-cutané est relié au nerf sacré S2 par l'intermédiaire d'une électrode bipolaire unique. Il ne dispose pas de batterie, mais d'une antenne bobinée lui permettant

de recevoir du contrôleur externe son énergie et ses données par le biais d'un lien inductif.

Donc, le NLB devra lui aussi être réalisé sur un circuit imprimé à partir de composants discrets et programmables, il devra être alimenté en énergie depuis l'extérieur, il devra générer un signal de stimulation sélective, il devra évaluer rapidement la résistance du nerf vue de son étage de sortie et il devra retourner la valeur mesurée à un afficheur externe via le même lien inductif.

Pour des raisons de simplicité, nous ne chercherons qu'à estimer le module de l'impédance à différentes fréquences, mais pas son argument. La plage de mesure devra s'étaler de 0Ω à $100K\Omega$, avec la meilleure résolution possible compte tenu des contraintes de design (100Ω serait acceptable).

1.3 Principe d'évaluation d'impédance

La caractérisation complète de l'impédance de l'interface nerf-électrodes est un sujet d'envergure, surtout lorsqu'elle implique l'utilisation d'un système implanté autonome, en mesure de s'autocalibrer lors de son initialisation et de déterminer automatiquement la gamme de capture [19], [20]. Nous allons donc nous limiter ici à décrire le principe de mesure simplifié qui est utilisé dans le NLB. Cette technique repose sur l'hypothèse que l'impédance est essentiellement résistive dans le cas particulier de notre application. En réalité, on observe plutôt un modèle série-parallèle à plusieurs composants passifs R et C.

Le microstimulateur injecte dans le nerf un courant I parfaitement contrôlé, qui va alors produire une différence de potentiel V directement proportionnelle à l'impédance

résistive du nerf (R). Mesurer V connaissant I revient à déterminer R ($V=R \cdot I$). De plus, pour faciliter l'intégration et améliorer la flexibilité, la tension V n'est pas directement quantifiée par un convertisseur analogique à numérique (CAN), mais elle est convertie en une fréquence par un VCO (Voltage Controlled Oscillator). Cette fréquence peut alors être facilement évaluée par un circuit numérique. Il est bien sûr nécessaire de connaître le coefficient de proportionnalité du VCO si celui-ci est linéaire, ou sinon de connaître sa courbe caractéristique V-F.

Les avantages de cette méthode sont plus significatifs dans le cas de la conception d'un circuit intégré dédié, où le design d'un VCO est plus aisé que celui d'un CAN. Dans notre cas, il s'agissait d'expérimenter le concept sur prototype avant son intégration, mais il a aussi permis de simplifier les interconnexions et de réduire le nombre de composants externes au FPGA.

1.4 Design du NLB

Contrairement à son prédécesseur, le NLB doit effectuer deux tâches distinctes. L'évaluation d'impédance ne sera pas effectuée durant la phase de stimulation sélective et inversement une mesure d'impédance ne devra pas provoquer de stimulation. Il existera donc une commande et des paramètres spécifiques pour chacun des modes de stimulation et de mesure d'impédance. La stimulation sélective fonctionnera de la même façon que pour le MNS, quand à la mesure d'impédance, elle se déroulera comme suit : Le nerf est stimulé par une onde carrée bipolaire de faible amplitude (inférieure au seuil d'activation de la contraction musculaire de la vessie), la tension résultante est évaluée pendant une durée T_{eval} et la valeur mesurée est retournée en boucle vers l'afficheur

externe. A chaque application du contrôleur externe, la mesure n'est faite qu'une seule fois, mais la donnée est transmise indéfiniment jusqu'à l'interruption de l'énergie RF.

La Figure 1.2 présente l'allure du signal de stimulation pour une mesure d'impédance, ainsi que les quatre paramètres programmables intervenant dans ce mode de fonctionnement. Ces paramètres sont les suivant :

- Amp (Amplitude) : amplitude du courant d'excitation
- Freq (frequency) : fréquence du courant d'excitation
- T_{setup} (setup time) : temps de stabilisation du VCO
- T_{eval} (evaluation time) : durée du comptage de la fréquence

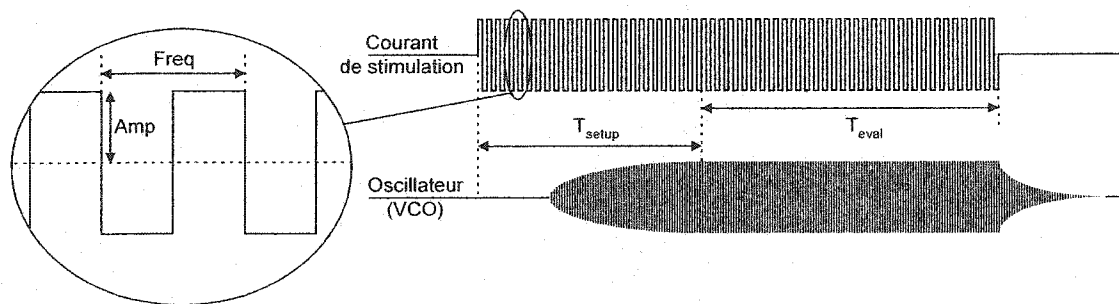


Figure 1.2 : Formes d'ondes des signaux pour la mesure d'impédance

1.4.1 Protocole de communication

Puisque nous devons avoir un nouveau stimulateur compatible avec le contrôleur externe déjà utilisé pour le MNS, nous avons du conserver entièrement la structure du protocole de communication existant. Ainsi, pour le mode de stimulation sélective, le fonctionnement et les paramètres sont parfaitement identiques à ceux du MNS, permettant une interchangeabilité parfaite du MNS avec le NLB. La Figure 1.3.a présente la structure d'une trame de données destinée à une stimulation sélective. Elle est

constituée d'une entête spécifique de 10 bits, d'un bit de synchronisation, d'un bit de départ et des six paramètres de la stimulation sélective étendus sur 52 bits.

Comme on peut le voir à la Figure 1.3.b, le protocole existant ne prévoyait pas de mot de commande, permettant de spécifier différents modes de fonctionnement et l'architecture adoptée ne se prêtait pas à rallonger la trame de données. Nous avons donc opté pour une seconde entête spécifique pour le mode d'évaluation d'impédance. De plus, seul quatre paramètres sont utiles pour la mesure d'impédance, il est donc nécessaire de rajouter quelques bits de remplissage pour avoir toujours une trame de 64bits. Enfin, nous verrons à la section 1.4.5 sur l'architecture du contrôleur sur FPGA qu'il a été possible de conserver la position et la longueur des paramètres dans la trame en réutilisant adéquatement la structure interne du FPGA.

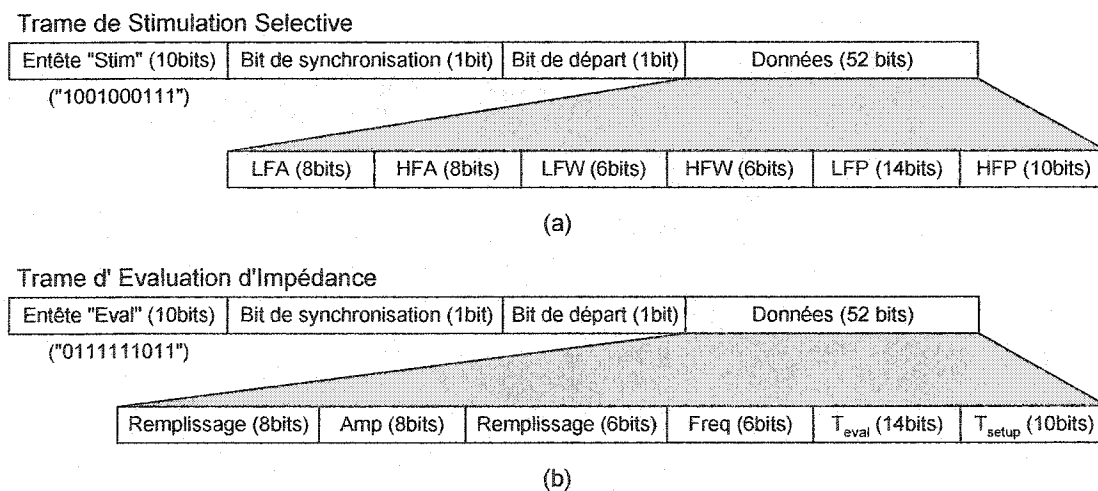


Figure 1.3 : Structures des trames de données du NLB

1.4.2 Architecture détaillée

Le schéma bloc de la Figure 1.4 présente les principaux blocs constitutifs du NLB. On remarque en particulier que le VCO utilisé pour la mesure d'impédance est

directement connecté à la même source de courant qui sert à la stimulation sélective. La mesure d'impédance n'a nécessité que l'ajout d'un VCO et de deux signaux de contrôle (*EN_VCO* et *OSC_IN*). De même, l'antenne bobinée est partagée entre le circuit de réception de l'énergie et des données et le circuit de transmission des mesures. Pour l'émission de données, l'implant se sert d'un signal FSK (Frequency Shift Keying) à 2MHz modulé à 4KHz. En laissant un décuple entre la porteuse de réception et la porteuse d'émission, on s'assure d'une bonne séparation des fréquences.

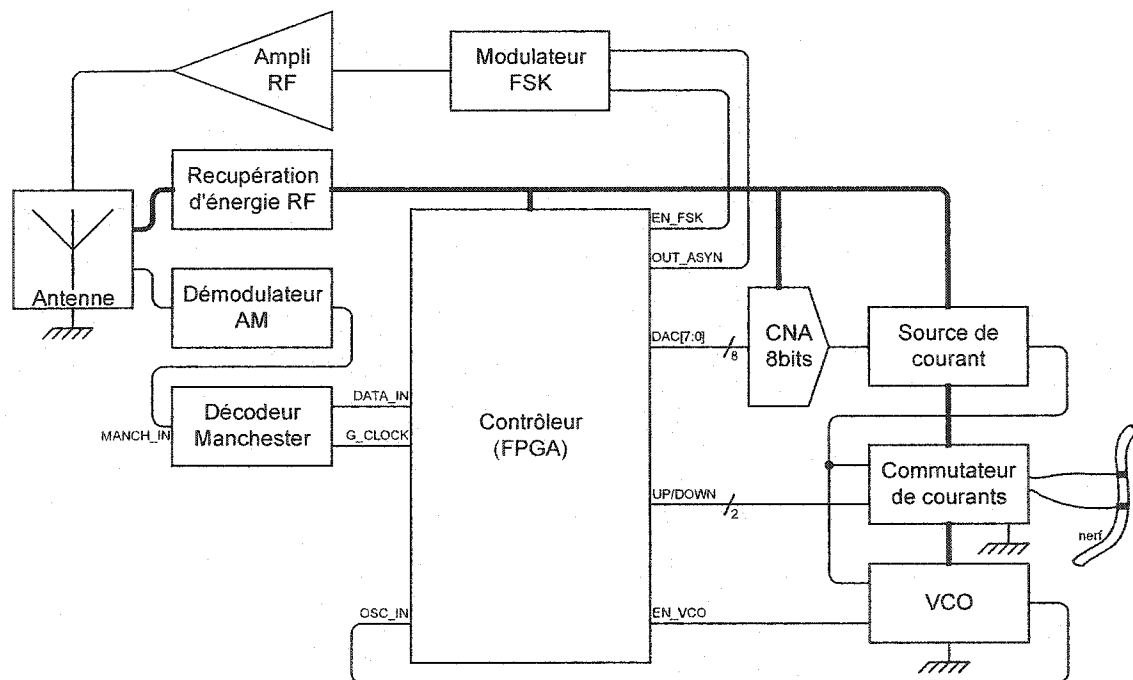


Figure 1.4 : Schéma bloc du NLB

Le signal RF de 20MHz modulé ASK (Amplitude Shift Keying) capté par l'antenne est régulé pour alimenter en énergie l'ensemble de l'implant. Démodulé, le signal binaire obtenu, qui est codé en format Manchester, est transmis au FPGA qui se charge alors d'en extraire l'horloge à 300KHz et les données à 300Kbps.

Un convertisseur numérique à analogique (CNA) commandé par le FPGA permet de contrôler l'amplitude du courant de stimulation, alors qu'un pont d'interrupteurs en H assure la bipolarité du courant appliqué au nerf.

Pour la mesure d'impédance, le VCO est contrôlé par la tension résultante à la sortie de la source de courant et le signal d'oscillation (*OSC_IN*) est directement transmis au FPGA, qui se charge alors d'en évaluer la fréquence. Le modulateur FSK, couplé à un amplificateur RF basse puissance, transmet sur l'antenne une valeur codée sur 8bits, correspondant à l'impédance de l'interface nerf-électrodes.

1.4.3 Étage de stimulation et de mesure

L'étage de sortie analogique connecté au nerf est responsable de la génération des courants bipolaires pour la stimulation sélective et la mesure d'impédance. De plus, un VCO permet de convertir la tension résultante à la sortie de la source de courant en une onde carrée de fréquence variable.

La partie génération de stimuli est constituée d'un CNA à 8bits, dont le niveau de la tension de sortie est ajusté par un amplificateur opérationnel (AOP). La tension mise à l'échelle est ensuite convertie en un courant par un autre AOP. Enfin, ce courant est injecté dans le nerf à travers un pont d'interrupteurs en H permettant d'inverser le sens du courant dans le nerf, sans avoir à changer le sens du courant au niveau de la source. Sur la Figure 1.5 présentant une partie de l'étage de sortie, on peut identifier la source de courant (AOP1 et R_i), le pont d'interrupteur (structure en H) et l'oscillateur à relaxation (VCO) constitué des AOP2, AOP3, transistor T et quelques éléments passifs.

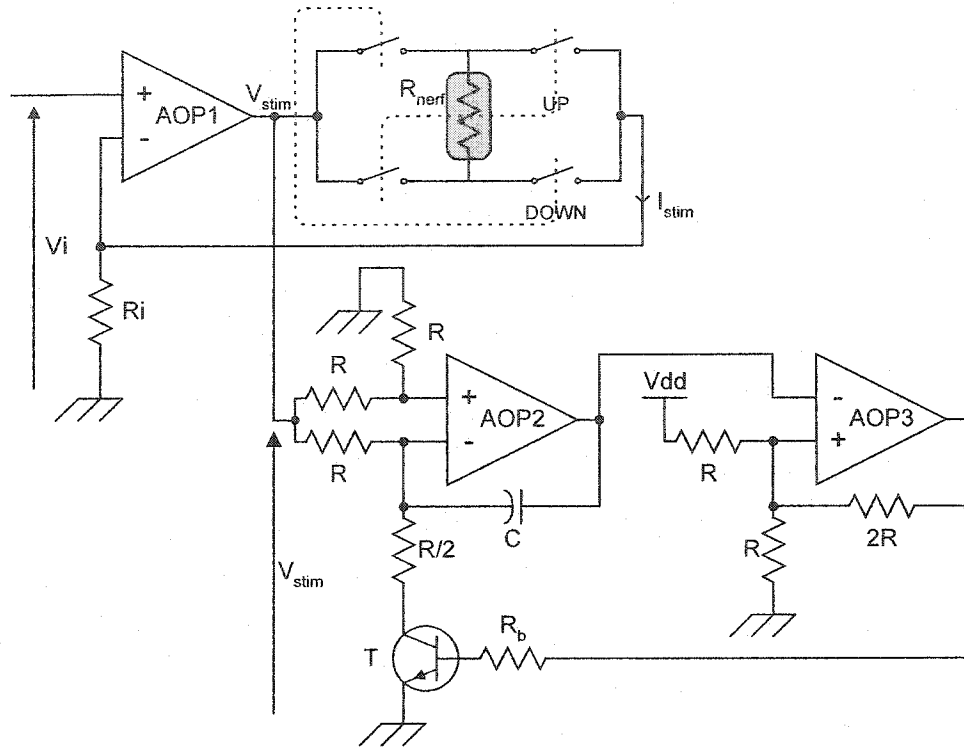


Figure 1.5 : Schéma électrique du VCO et de la source de courant

La valeur du courant de stimulation est donc $I_{stim} = V_i / R_i$. Il faut cependant choisir R_i en s'assurant que la tension résultante V_{stim} à la sortie de l'AOP1 reste au dessous de la tension de saturation positive (V_{sat+}) de l'AOP (1). La résistance R_i doit donc être calculée à partir de l'équation (2).

$$V_{sat+} = R_{nerf} * I_{stim(max)} + V_i \quad (1)$$

$$R_i = V_{sat+} / I_{stim(max)} - R_{nerf} \quad (2)$$

Le VCO utilisé pour la conversion de la tension V_{stim} en une fréquence est un oscillateur à relaxation, où le condensateur C est alternativement chargé puis déchargé à courant constant par l'AOP2 (en supposant que le transistor T est un commutateur idéal).

Lors de la charge du condensateur C, le transistor T conduit et le courant de charge I_{charge} est donné par la relation (3).

$$I_{\text{charge}} = (V_{\text{stim}} / R) - (V_{\text{stim}} / 2.R) = V_{\text{stim}} / 2.R \quad (3)$$

La tension à la sortie de l'AOP2 augmente jusqu'à atteindre la tension de seuil supérieur (V_+) du comparateur à hystérésis réalisé avec l'AOP3. Le transistor T est alors bloqué et le condensateur C se décharge à courant constant $I_{\text{décharge}}$.

$$I_{\text{décharge}} = V_{\text{stim}} / 2.R \quad (4)$$

Les seuils du comparateur sont donnés par les équations (5) et (6).

$$V_+ = \frac{R}{R + R//2R} V_{\text{dd}} = \frac{3}{5} V_{\text{dd}} \quad (5)$$

$$V_- = \frac{R//2R}{R + R//2R} V_{\text{dd}} = \frac{2}{5} V_{\text{dd}} \quad (6)$$

La fréquence d'oscillation (F_{osc}) est donc directement proportionnelle à V_{stim} .

$$F_{\text{osc}} = \frac{I}{2 * C * \Delta V} = \frac{V_{\text{stim}}}{4 * RC * \Delta V} = \frac{V_{\text{stim}}}{4/5 * C * V_{\text{dd}}} \quad (7)$$

L'onde carrée destinée à être évaluée par le FPGA est alors disponible à la sortie de l'AOP3.

En pratique, le transistor T ne fonctionnant pas en mode de saturation, il introduira une légère non-linéarité dans le VCO, car les courants de charge et de décharge du condensateur C ne seront plus constants.

1.4.4 Lien RF bidirectionnel

La partie analogique de l'étage de réception, responsable de la récupération de l'énergie et des données, est basée sur un circuit à résonance parallèle. Ce dernier est

constitué de l'antenne bobinée et d'un condensateur. Le courant alternatif induit est redressé à double alternance par deux diodes puis filtré par un condensateur de forte capacité. La régulation est assurée par une diode zener à seuil ajustable. La démodulation du signal ASK, utilise un détecteur d'enveloppe classique ne comportant qu'un transistor et une résistance. Les schémas électriques de ces circuits sont présentés sur la Figure A.1.

La seconde fonction de l'étage RF est de transmettre vers un récepteur externe les données mesurées à l'interne (la valeur de l'impédance). Pour pouvoir réutiliser l'antenne de réception comme moyen de transport des données mesurées, nous avons choisi un oscillateur FSK contrôlé directement par le FPGA et connecté à l'antenne par un circuit d'adaptation d'impédance LC. La fréquence centrale est de 2MHz, la déviation maximale de 100KHz et le taux de modulation de 4KHz.

L'oscillateur FSK de la Figure 1.6.a est un oscillateur Colpitt à éléments discrets [71]. En l'absence du condensateur C, la fréquence est donnée par l'équation (8).

$$F_0 = \frac{1}{2\pi\sqrt{L\left(\frac{1}{C_1} + \frac{1}{C_2}\right)^{-1}}} \quad (8)$$

Rajouter C, revient à introduire un nouveau condensateur en série avec les condensateurs C1 et C2. La nouvelle fréquence de résonance est alors donnée par l'équation (9).

$$F_i = \frac{1}{2\pi\sqrt{L\left(\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C}\right)^{-1}}} \quad (9)$$

On voit donc qu'il est possible d'obtenir deux fréquences d'oscillation distinctes en activant ou en désactivant le condensateur C. Cette tâche est assurée par une des

sorties du FPGA qui est configurée en drain ouvert et court-circuite le condensateur C pour le "retirer" du circuit résonnant.

Le modulateur FSK est suivi d'un amplificateur classe A à basse puissance qui permet uniquement d'isoler l'oscillateur du reste du circuit d'émission.

Le circuit d'adaptation d'impédance présenté à la Figure 1.6.b permet de brancher le modulateur FSK au système de réception des données et de l'énergie sans en perturber le fonctionnement. En effet, il ne faut surtout pas que la présence du circuit de transmission à 2MHz vienne changer la fréquence et la qualité du circuit résonnant à 20MHz. De plus, l'adaptation d'impédance est là pour augmenter l'impédance de l'antenne vue de l'amplificateur de sortie. Compte tenu de la proximité des deux antennes lors de la communication, nous nous sommes permis de réduire le facteur de qualité de l'antenne (pour les signaux à 2MHz) en ajoutant la résistance R_L . Ainsi, l'adaptation d'impédance devenait plus aisée et le canal moins sélectif.

Le schéma électrique de tout l'étage de réception-transmission est disponible sur la Figure A.1.

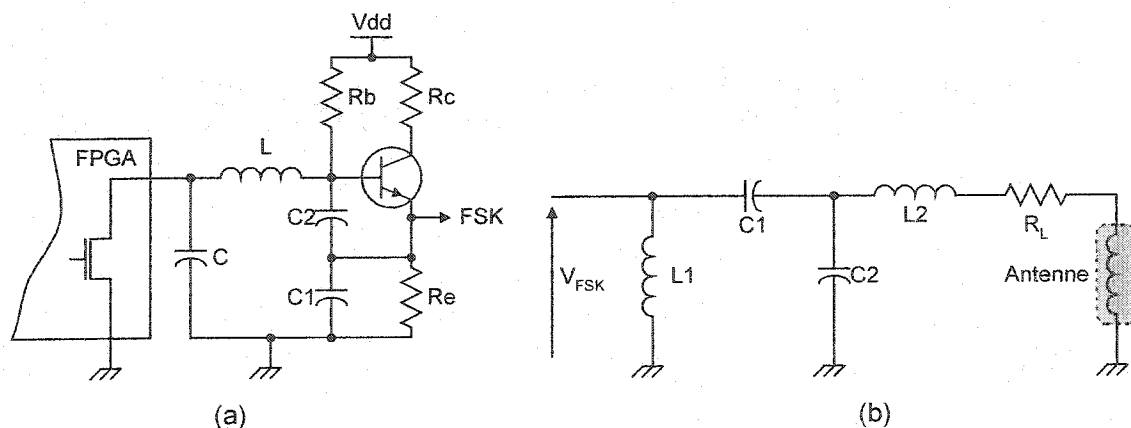


Figure 1.6 : Schéma électrique de l'oscillateur FSK et de l'adaptateur d'impédance

1.4.5 Contrôleur intégré sur FPGA

La première tâche à effectuer par le contrôleur du NLB est de décoder le signal binaire reçu en format Manchester du contrôleur externe. Notre système utilise une configuration à trois bascules D et un circuit de délai de $\frac{3}{4}T$. La Figure 1.7 présente la structure d'un tel décodeur où les deux premières bascules détectent les transitions du signal et le délai est assuré par une cellule RC. La diode permet d'obtenir une forme d'onde en dents de scie au niveau de la commande de réinitialisation des bascules.

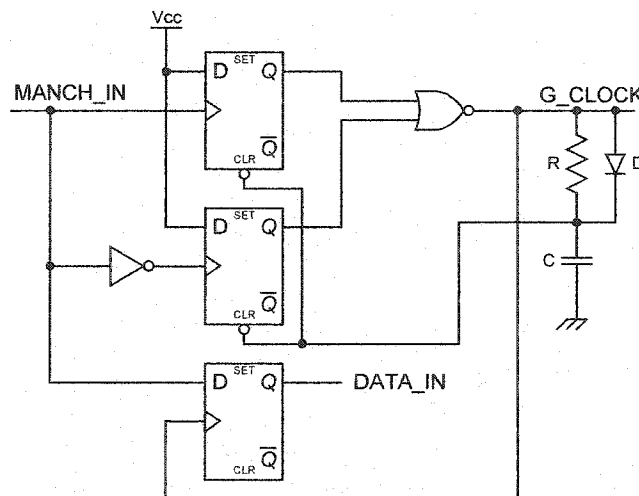


Figure 1.7 : Schéma du décodeur Manchester

Le décodage de ce signal Manchester génère l'horloge globale du système à 300KHz (G_CLOCK) et récupère la trame de données de 64 bits. Ces bits sont décalés un à un dans une file à détection d'entête de 53 bits. Dans un premier temps, seuls les 10 premiers bits du registre sont utilisés, ce qui permet au système de détecter l'arrivée de l'entête valide de 10 bits au début d'une trame de 64 bits. Une fois l'entête détectée, le registre au complet est réinitialisé le temps d'un cycle d'horloge, le onzième bit de la trame est sacrifié, puis débute l'acquisition des 53 bits de données restant. Il est à noter

que le douzième bit d'une trame (toujours égal à "1") est dédié à signaler l'arrêt du décalage des 52 bits de données utiles dans le registre.

a) Génération de stimuli pour la stimulation sélective

Une fois les paramètres disponibles sur le bus de données interne du FPGA, le contrôleur peut commencer la génération des stimuli ou la mesure d'impédance en fonction de l'entête spécifique qu'il a reçu au début de la trame.

La Figure 1.8 présente les différents blocs constitutifs du contrôleur interne du NLB intégré sur FPGA.

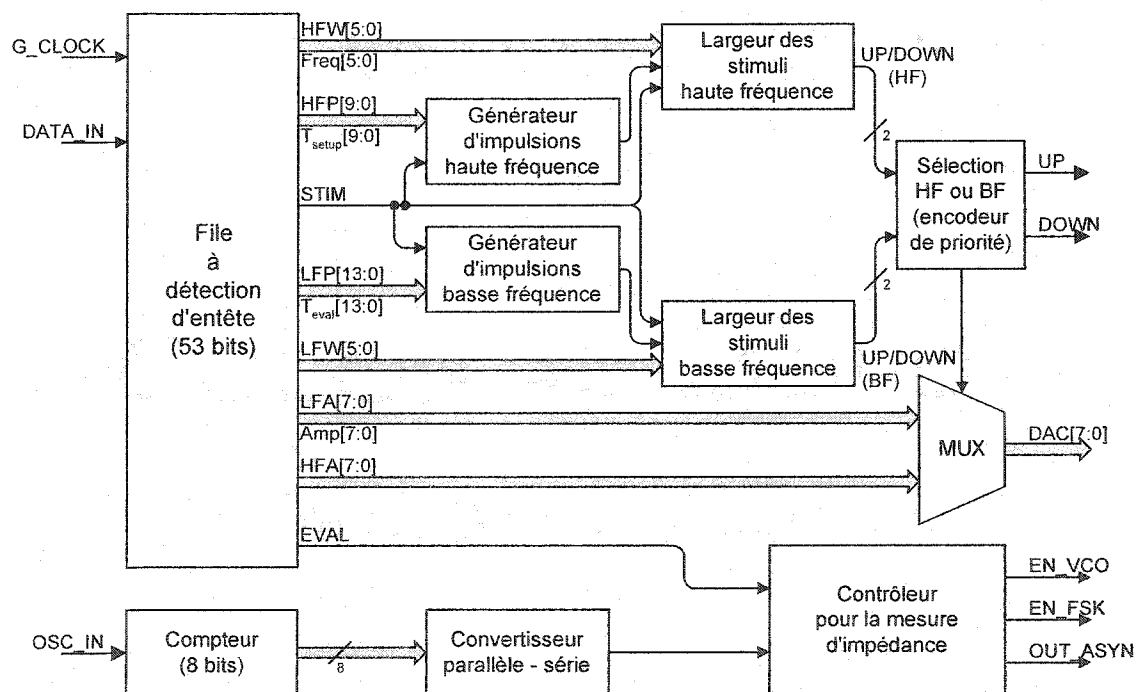


Figure 1.8 : Schéma bloc du contrôleur sur FPGA du NLB

Lorsqu'une entête de stimulation valide est détectée, les deux générateurs d'impulsions haute et basse fréquences seront activés. Ces générateurs procèdent à la division du signal d'horloge de 300KHz (G_CLOCK) par les nombres entiers reçus dans

la trame de données ($HFP[9:0]$ et $LFP[13:0]$). Les impulsions à basse et haute fréquences générées déclenchent ensuite leur contrôleur de largeur de stimuli respectifs. Ces deux contrôleurs indépendants sont responsables de commander les signaux de contrôle du commutateur de courant (UP et $DOWN$) à intervalle régulier spécifié par les données $HFW[5:0]$ et $LFW[5:0]$. Les temporisations internes sont assurées par des diviseurs de fréquence. Le dernier module est un encodeur de priorité qui combine les signaux UP et $DOWN$ aux deux fréquences et sélectionne l'amplitude de stimulation correspondante ($HFA[7:0]$ et $LFA[7:0]$) à transmettre au CNA. La priorité est donnée aux signaux de contrôle à basse fréquence. Le Tableau 1.2 établit la correspondance entre les mots d'une trame et les paramètres de stimulation.

Tableau 1.2 : Les paramètres d'une trame et leur expression littérale

Acronyme	Longueur	Expression littérale	Valeur min	Valeur max	Résolution ¹
HFP	10 bits	$(HFP[9:0] * T_{cy})^{-1}$	293 Hz	2000 Hz	1.2 Hz
T_{setup}	10bits	$T_{setup}[9:0] * T_{cy}$	0 μ s	3.4 ms	3.33 μ s
LFP	14 bits	$(LFP[13:0] * T_{cy})^{-1}$	18 Hz	2000 Hz	3 mHz
T_{eval}	14 bits	$T_{eval}[13:0] * T_{cy}$	0 μ s	54.6 ms	3.33 μ s
HFW	6 bits	$HFW[5:0] * T_{cy}$	0 μ s	210 μ s	3.33 μ s
Freq	6 bits	$(Freq[5:0] * T_{cy})^{-1}$	2.3 KHz	150 KHz	100Hz
LFW	6 bits	$LFW[5:0] * T_{cy}$	0 μ s	210 μ s	3.33 μ s
HFA	8 bits	$(HFA[7:0]/255) * I_{max}$	0 μ A	2 mA	8 μ A
Amp	8 bits	$(Amp[7:0]/255) * I_{max}$	0 μ A	2 mA	8 μ A
LFA	8 bits	$(LFA[7:0]/255) * I_{max}$	0 μ A	2 mA	8 μ A
I_{max} = Courant maximal de sortie (2 mA) T_{cy} = Durée d'un cycle d'horloge (3.33 μ s) ¹ Résolution autour d'une valeur typique (0.9mA, 30Hz, 175 μ s, 1.3mA, 600Hz, 100 μ s)					

b) Évaluation de l'impédance

Pour la mesure d'impédance, le nerf est stimulé par un courant alternatif (onde carrée) puis la tension résultante est évaluée pendant une durée T_{eval} . De plus, il est nécessaire d'attendre la stabilisation en fréquence du VCO pendant une durée T_{setup} avant

de déclencher le compteur de fréquence. Ainsi, le circuit de mesure regroupe un registre pour l'amplitude de stimulation, un contrôleur du commutateur de courant et deux temporisateurs (timer). Ces quatre derniers éléments sont déjà présents dans l'étage de sortie du stimulateur, il suffit donc de les partager entre les deux fonctions du NLB et d'ajouter un contrôleur qui en supervisera le fonctionnement.

Comme on peut le voir sur la Figure 1.8 et dans le Tableau 1.2, les générateurs d'impulsions haute et basse fréquence ont été utilisés comme temporisateur pour les durées T_{setup} et T_{eval} . De même le contrôleur de largeur d'impulsion haute fréquence ainsi que le paramètre d'amplitude qui lui est associé serviront à générer le courant alternatif de stimulation. Le module de contrôle basse fréquence, quant à lui, demeure inactif dans ce mode. Enfin, le contrôleur pour la mesure d'impédance est une machine à états qui supervise la séquence d'exécution des étapes intervenant dans la mesure d'impédance. Ces étapes se résument comme suit :

- activer la stimulation et le VCO
- attendre la stabilisation du VCO pendant une durée T_{setup}
- compter les périodes du signal issu du VCO pendant une durée T_{eval}
- arrêter la stimulation et le VCO
- activer le modulateur FSK
- transmettre via le modulateur FSK la donnée de 8 bits selon un protocole standard sériel asynchrone à 1 bit de départ, 8 bits de données, 1 bit de parité paire et 2 bits d'arrêt.

Les schémas logiques et les codes VHDL du contrôleur sur FPGA du NLB sont présentés à l'annexe A.

1.5 Réalisation du prototype

Un prototype du NLB a été réalisé sur un circuit imprimé (PCB) avec un circuit intégré (CI) programmable et des composants discrets. Un FPGA 40MX04 d'Actel a été utilisé pour intégrer toute la partie numérique de l'implant. Les composants analogiques sont, quant à eux, empruntés en bonne partie de la version précédente du stimulateur (MNS). Le VCO par exemple, utilise un CI à double AOP plutôt qu'une combinaison AOP-comparateur qui aurait été préférable. De même, l'oscillateur FSK et tout l'étage d'émission a nécessité beaucoup de composants discrets, car il n'était pas possible de trouver des CI commerciaux à faible puissance qui remplissent les fonctions souhaitées.

L'antenne a été intégrée au PCB, par un enroulement concentrique de pistes de cuivre sur la périphérie du substrat circulaire. Les performances réelles d'une telle antenne sont tout à fait comparables à celles d'une antenne en fils de cuivre bobiné.

Bien sûr, le système a nécessité quelques réajustements dans la valeur de certains éléments, en particulier au niveau de l'oscillateur FSK et du circuit d'adaptation d'impédance de l'antenne, mais le comportement du PCB était tout à fait satisfaisant compte tenu de l'absence de plan de masse et de la concentration des composants. La Figure 1.9 présente une photographie à l'échelle 1/1 des 2 faces du prototype du NLB.

Les schémas électriques et logiques du NLB sont présentés à l'annexe A. Les dessins des masques du prototype réalisé se trouvent dans cette même annexe.

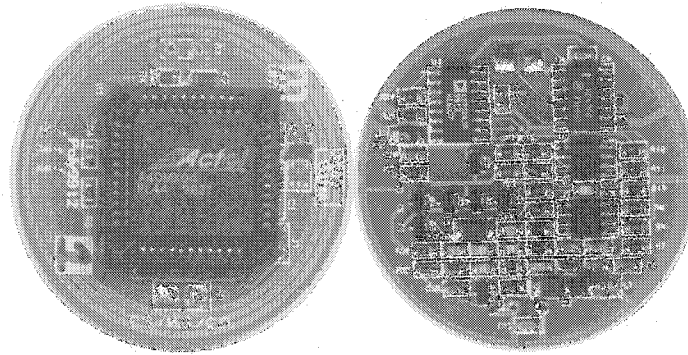


Figure 1.9 : Photographie du stimulateur NLB à l'échelle 1/1

Ce prototype a été complété et a fait l'objet d'une caractérisation en laboratoire. Nous verrons au chapitre 4 les performances et les limites de ce système, qui a répondu convenablement à l'objectif d'ajouter un module d'évaluation de l'impédance du nerf tout en étant entièrement compatible avec la version antérieure du stimulateur.

1.6 Conclusion

Le suivi in vivo de l'état de l'interface nerf-électrodes est une information très utile d'un point de vue médical dans la réhabilitation des fonctions vésicales, mais aussi d'un point de vue ingénierie dans le design de l'étage de stimulation. En effet, les résultats d'expérimentations en phase chronique chez l'animal présentés au chapitre 4 nous permettront de constater l'importance de surveiller cette impédance afin d'améliorer la fiabilité des stimulateurs implantables.

Il est à noter que les tests in vivo du NLB n'ont pas eu lieu, nous avons plutôt privilégié la mise au point d'un nouveau système de stimulation en mesure d'effectuer à la fois une stimulation sélective sur demande et une stimulation permanente à faible amplitude. La conception, la réalisation et les tests de cette nouvelle génération de microstimulateur font l'objet du reste de ce mémoire.

CHAPITRE 2

NEUROSTIMULATEUR SÉLECTIF ET PERMANENT

2.1 *Introduction*

Lorsque le lien entre le cerveau et la vessie est interrompu suite à une lésion de la moelle épinière, les contractions réflexes de la vessie, qui peuvent survenir lors de son remplissage, ne sont plus inhibées par le cerveau et s'amplifient avec le temps. Cette phase d'hyperactivité des contractions réflexes de la vessie, appelée hyper-réflexie, apparaît quelque mois après le traumatisme spinal. Une stimulation électrique continue, à longueur de journée, à basse fréquence et à faible amplitude devrait permettre de rétablir cette régulation en simulant en quelque sorte la présence du cerveau. Ainsi, nous proposons dans ce chapitre un système électronique implantable capable d'effectuer une telle stimulation. Ce nouveau neurostimulateur sélectif et permanent (NSP) assurera 2 types de stimulation, une stimulation sélective par blocage haute fréquence et une stimulation permanente à faible amplitude. Les prototypes utiliseront des circuits intégrés disponibles commercialement et seront conçus pour permettre des implantations réelles chez l'animal.

2.2 *Spécifications*

Tout comme ses prédécesseurs, ce microstimulateur doit pouvoir générer ponctuellement un signal de stimulation sélective à partir des données transmises par un contrôleur externe. L'énergie nécessaire à la génération de ces stimuli proviendra d'un

couplage inductif entre le contrôleur externe et l'implant sous cutané. La forme d'onde d'un signal de stimulation sélective par blocage haute fréquence, ainsi que les 6 paramètres programmables, ont déjà été présentés à la section 1.2.

En plus de la stimulation sélective, le nouveau microstimulateur doit pouvoir générer de façon autonome et continue un signal dit de stimulation permanente. Ce signal, similaire au signal basse fréquence de la stimulation sélective, devra être un courant bipolaire, de faible amplitude, pulsé à basse fréquence en train d'impulsions de durée programmable. La Figure 2.1 représente l'allure souhaitée d'un signal de stimulation permanente avec ses 5 paramètres programmables.

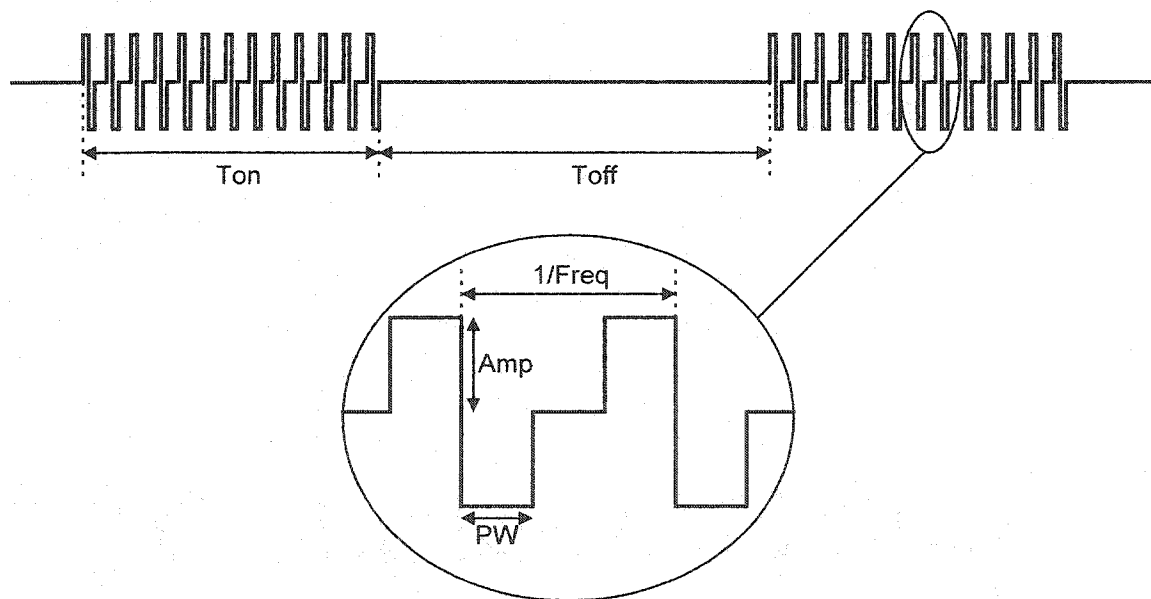


Figure 2.1 : Forme d'onde d'un signal de stimulation permanente

Bien que le système doit être autonome pour la génération des stimuli, il n'en restera pas moins programmable à distance par le même contrôleur externe utilisé pour la stimulation sélective. Le Tableau 2.1 contient la description des 5 paramètres

programmables pour un signal de stimulation permanente. Les valeurs de plage et de résolution indiquées dans ce tableau sont les performances nominales souhaitées.

Tableau 2.1 : Description des paramètres pour la stimulation permanente

Paramètre	Amplitude	Fréquence (1/période) ¹	Largeur d'impulsion	Durée active	Durée inactive
Acronyme	Amp	Freq	PW	Ton	Toff
Unité	μA	Hz	μs	sec	sec
Plage	0 - 1000	5 - 100	10 - 300	10 - 60	1 - 10
Résolution	10	1	10	5	1
Valeur typique	200	30	175	20	10
¹ Le paramètre Freq correspond à la fréquence des stimuli de courant c-à-d l'inverse de la période qui sépare 2 stimuli					

Puisque le système doit effectuer une stimulation continue même en l'absence du contrôleur externe, il ne pourra plus disposer de l'énergie inductive et devra donc posséder sa propre source d'énergie, soit une pile. Donc, le nouveau microstimulateur aura 2 sources d'alimentation en énergie qui devront être adéquatement gérées pour maximiser la durée de vie de la source non renouvelable. Dans le cadre de nos expériences in vivo, une durée de vie minimale de 8 mois sera suffisante.

Enfin, il est essentiel de prévoir un moyen d'activer et de désactiver à distance la fonction de stimulation permanente. Il est aussi souhaitable de pouvoir mettre hors tension l'ensemble de l'implant à distance dans le cas d'un stockage de longue durée ou d'une interruption prolongée du traitement.

2.3 Architectures envisagées

Les nouveaux besoins plus contraignants de cette troisième génération de microstimulateurs ne permettent pas, comme dans le cas de la version précédente, de conserver intégralement l'architecture initiale (MNS) et d'y greffer les nouvelles

fonctionnalités. L'ajout d'une seconde source d'énergie nous a amené à envisager différentes méthodes de gestion d'énergie et de partage des ressources. Il a fallu trouver un compromis entre la réutilisation de structures déjà existantes, la fiabilité de fonctionnement et la simplicité d'utilisation par le patient.

La Figure 2.2 illustre les 4 principales architectures envisagées pour le nouveau microstimulateur. La première (a) supposait une réutilisation du FPGA (Field Programmable Gate Array) comme contrôleur central pour les 2 types de stimulation. Un sélecteur, placé sur la ligne d'alimentation du FPGA et de l'étage de sortie, permettait de commuter entre les 2 sources d'énergie. Il était alors difficile de garantir une bonne commutation des alimentations sans risquer de perdre les données mémorisées pour la stimulation permanente lorsque la source d'énergie inductive était retirée. La seconde proposition (b) voulait régler le problème de la discontinuité de l'alimentation du FPGA en connectant ce dernier sur la pile de façon permanente. La commutation des sources d'énergie ne s'appliquait que pour l'étage de sortie. Cependant, les performances énergétiques du FPGA dont nous disposions n'étaient pas adaptées pour un système qui se devait d'être à très basse consommation. Est alors venu l'idée de changer le FPGA du contrôleur central, par un microcontrôleur (c) qui avait l'avantage de disposer d'une mémoire non volatile et de fonctions d'économie d'énergie avancées. Toutefois, pour parvenir aux mêmes performances en stimulation sélective qu'avec le FPGA, il aurait fallu accroître la fréquence de fonctionnement interne. Ainsi les avantages d'économie d'énergie de ce type de contrôleur n'aurait plus été aussi évident.

Finalement, une architecture utilisant les 2 types de contrôleurs (d) permettrait de tirer avantage de chacun d'entre eux. D'où la décision finale d'utiliser un FPGA alimenté par l'énergie inductive pour la fonction de stimulation sélective et un microcontrôleur (PIC) alimenté par pile pour la fonction de stimulation permanente. Les 2 contrôleurs se partageant le même étage de sortie.

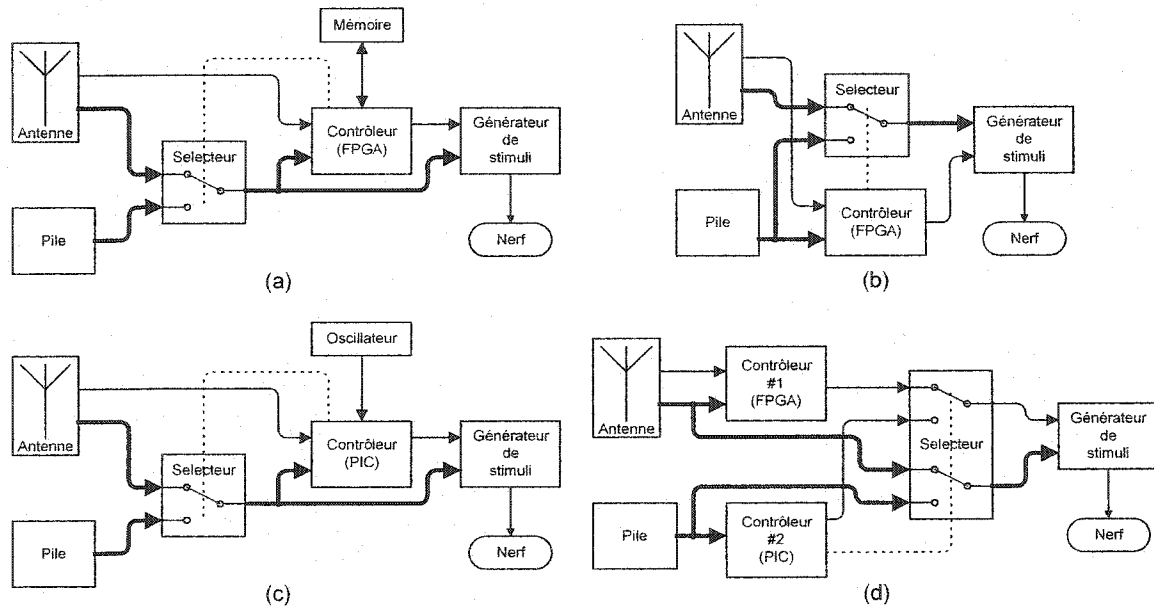


Figure 2.2 : Propositions d'architectures pour le NSP

2.4 Design du NSP

Dans cette architecture à 2 générateurs de stimuli indépendants, seul le FPGA s'alimente à partir du lien inductif RF et est responsable de la récupération et du décodage de l'ensemble des données transmises par le contrôleur externe à l'implant. Le PIC doit par conséquent récolter ses données, partiellement décodées, via le FPGA et est uniquement alimenté par la pile. La Figure 2.3 illustre le cheminement des données et de

l'énergie dans le microstimulateur ainsi que le partage des ressources au niveau de l'étage de sortie.

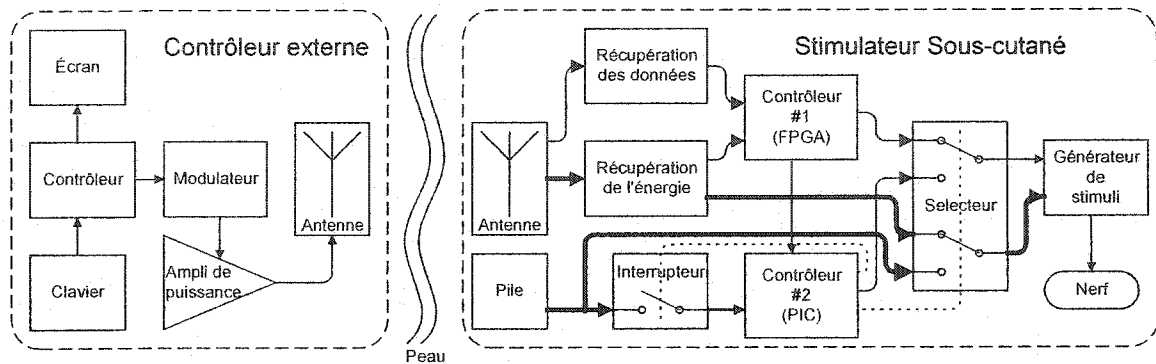


Figure 2.3 : Schéma bloc simplifié du système de stimulation NSP

Dans le cas d'une stimulation sélective, le contrôleur externe fournit au microstimulateur sous-cutané l'énergie et les paramètres nécessaires pour la génération des stimuli. Le FPGA prend le contrôle de l'étage de sortie qui est alors alimenté par l'énergie RF. Pour une stimulation permanente, le contrôleur externe est inutile car l'implant dispose alors d'une pile et le PIC utilise des paramètres de stimulation pré-programmés.

2.4.1 Modes de fonctionnement

La commutation entre la stimulation sélective et la stimulation permanente nécessite d'introduire la notion de modes d'opération. Nous avons ainsi défini 5 modes dont chacun représente un état stable de fonctionnement du microstimulateur. Ces modes sont définis ci-dessous :

- **Éteint** : état par défaut du microstimulateur, où tout le système est hors tension.

Dans ce mode, l'implant est complètement inerte pour son milieu environnant et la pile

ne se décharge pas. Par la suite, l'implant ne peut être allumé que par le contrôleur externe.

- **Sommeil** : état d'économie d'énergie avancée du système. Dans ce mode, l'implant est encore complètement inerte pour son milieu environnant et ne consomme que très peu d'énergie (5%). Il permet cependant d'être réveillé à tout moment soit par le contrôleur externe soit par un temporisateur interne. C'est un mode qui peut être utilisé par le patient à la place du mode *Éteint*, mais il est surtout utile pour les périodes de non stimulation (Toff) dans le mode de stimulation permanente.

- **Stimulation sélective** : état dans lequel le microstimulateur génère la combinaison d'impulsions haute et basse fréquences, qui constituent une onde dite de stimulation sélective, permettant la miction volontaire du patient. L'implant sortira de ce mode dès que le contrôleur externe sera retiré.

- **Stimulation permanente** : état dans lequel le microstimulateur génère de façon autonome des trains d'impulsions à basse amplitude, à basse fréquence et à intervalles réguliers pour constituer une onde dite de stimulation permanente. Ce mode n'est possible qu'en l'absence d'alimentation du contrôleur externe. L'implant entre et sort de ce mode uniquement sur commande du contrôleur externe.

- **Programmation** : état transitoire pendant lequel le contrôleur externe transmet à l'implant les nouveaux paramètres de stimulation permanente ou les commandes de commutation de modes.

2.4.2 Protocole de communication

Puisque le PIC ne peut communiquer directement avec le contrôleur externe, il a fallu mettre au point un protocole de communication utilisant la technique d'encapsulation de données. Ainsi le microstimulateur peut recevoir 3 types de trames.

La Figure 2.4 présente les 3 différentes trames possibles. La première (a) est un cas particulier d'une trame courte permettant simplement de ré-initialiser le système de récupération des données avant l'envoi d'une trame utile. Les 2 autres trames (b) et (c) présentent une structure classique avec un mot d'entête, un mot de commande, un mot de données et un mot de signature (code redundancy check, CRC) pour la détection d'erreurs [70]. L'entête identifie le début de la trame et le CRC permet de détecter des erreurs dans le mot de commande et le mot de données. Le mot de commande permet de préciser si les données sont destinées à la stimulation sélective (i.e. au FPGA) ou à la programmation de la stimulation permanente (i.e. au PIC). Si les données sont destinées à la programmation de la stimulation permanente (c), le mot de données est alors sous divisé en un mot de commande, un mot de données et un mot de signature (Checksum). La nouvelle signature est calculée sur le nouveau mot de commande et le nouveau mot de données.

Le Tableau 2.2 rassemble les codes des mots de commandes et les principales règles adoptées pour le protocole de communication du contrôleur externe vers l'implant.

Afin d'éviter que l'entête "0111 1110" ne se retrouve dans les données, il est nécessaire de rajouter dans la trame le bit de remplissage "0" après chaque séquence de cinq "1" consécutifs au niveau de l'émetteur. Ces "0" de bourrage seront ignorés par le

récepteur qui reconstituera la trame de 76 bits exactement. Cette technique est évidemment inspirée du protocole standard de communication X25 [70].

(a) Trame de Ré-initialisation

Entête (8bits)	Commande FPGA de ré-initialisation (4bits)
----------------	--

(b) Trame de Stimulation Selective

Entête (8bits)	Commande FPGA (8bits)	Données FPGA (52 bits)	CRC (8bits)		
LFA (8bits)	HFA (8bits)	LFW (6bits)	HFW (6bits)	LFP (14bits)	HFP (10bits)

(c) Trame de Stimulation Permanente

Entête (8bits)	Commande FPGA (8bits)	Données FPGA (52 bits)	CRC (8bits)	
Commande PIC (8bits)	Données PIC (32 bits)	Checksum (8bits)	Remplissage (4bits)	
Amp (8bits)	PW (8bits)	Freq (8bits)	Ton (4bits)	Toff (4bits)

Figure 2.4 : Structures des trames de données du NSP

Tableau 2.2 : Codes et conventions du protocole de communication avec le NSP

Mot de la trame	Valeur	Remarque
Entête	"0111 1110"	
Commande de ré-initialisation	"0101"	
Commande FPGA	"1001 1001" "1010 1010"	Stimulation sélective Nouveau paramètres de stimulation permanente
Commande PIC	"1110 1110" "1101 1101" "1011 1011" "0111 0111"	Éteindre Sommeil Nouveaux paramètres Activer la Stimulation
CRC	Polynôme X^8+1	Calculé sur Commande FPGA + Données FPGA
Checksum	Somme 8bits modulo 256	Calculé sur Commande PIC + Données PIC
Remplissage	"0000"	Pour avoir toujours des trames de 76 bits

2.4.3 Architecture détaillée

Comme on peut le voir à la Figure 2.5, c'est le FPGA qui est en charge de récupérer et décoder les données transmises par le contrôleur externe. Le signal reçu à l'antenne est un signal RF de 20MHz modulé en AM à un taux de 600Kbps. Un circuit analogique permet à la fois de réguler ce signal pour alimenter la majeure partie de l'implant en énergie et de le démoduler avant de le transmettre au FPGA. Ce dernier reconstitue alors l'horloge (300KHz) et les données (300Kbps) en décodant un signal préalablement codé en Manchester au niveau du contrôleur externe.

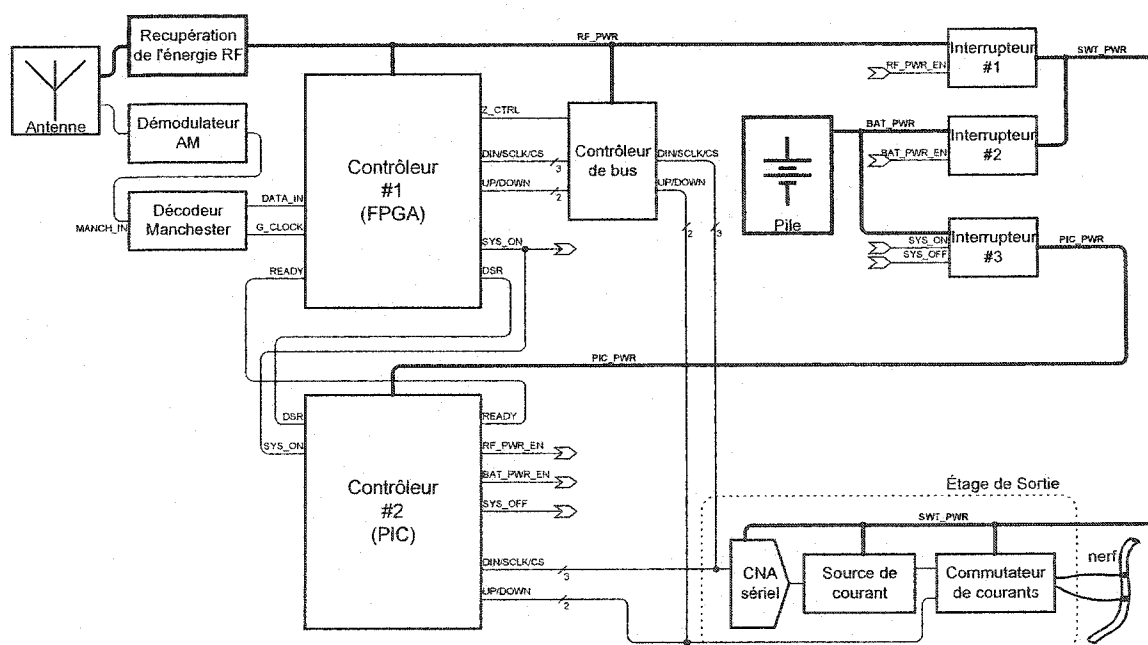


Figure 2.5 : Schéma bloc détaillé du NSP

De plus, la position privilégiée du PIC, qui est alimenté par une source d'énergie permanente, le prédispose à être le responsable de la commutation des sources d'alimentation de l'étage de sortie. De même il est indirectement l'arbitre du bus de données (signaux *CS/SCLK/DIN* et *UP/DOWN*) reliant les 3 principaux blocs : le FPGA,

le PIC et l'étage de sortie. Le contrôleur de bus permet de placer les sorties du FPGA en haute impédance lorsque celui-ci n'est pas alimenté en énergie.

Pour illustrer les techniques de commutation d'un mode à l'autre et la gestion des sources d'énergie, nous présentons à la Figure 2.6 un diagramme d'état simplifié du fonctionnement de l'ensemble du microstimulateur. Sur ce diagramme nous avons pris l'exemple du mode *Éteint* comme mode initial, mais il est important de noter que les modes *Sommeil* et *Stimulation permanente* peuvent aussi être des états initiaux. Ainsi sur le diagramme de la Figure 2.6, les états *Sommeil* et *Stimulation permanente* ne sont pas des états sans suite. Deux autres diagrammes identiques pourraient être dessinés avec les états *Sommeil* et *Stimulation permanente* comme états initiaux.

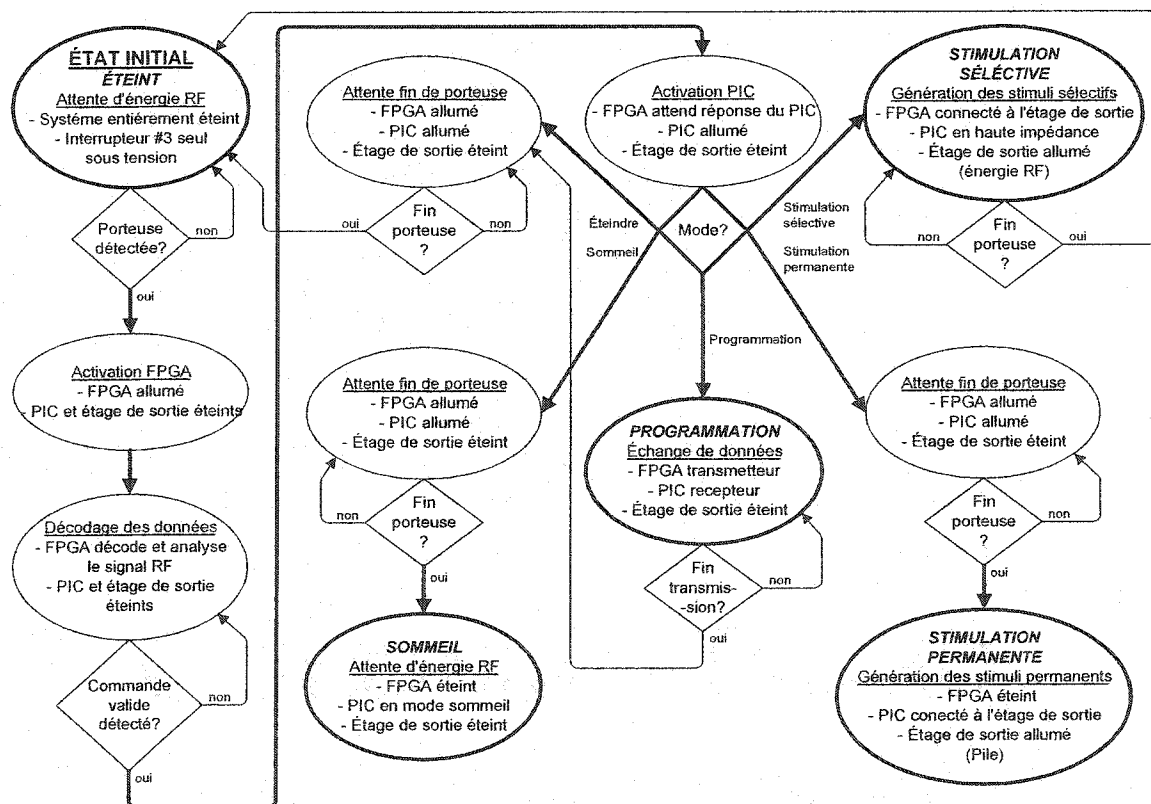


Figure 2.6 : Diagramme d'états simplifié du NSP

Dans tous les cas, si la porteuse est interrompue avant qu'une commande valide n'ait été exécutée, l'implant retournera automatiquement dans son état initial. On entend ici par état initial, l'état dans lequel se trouvait l'implant avant la détection de la porteuse à 20 MHz. C'est à partir des nouveaux paramètres de stimulation permanente que le système changera d'état initial en fonction du nouveau mode d'opération à activer, ce dernier étant précisé par le mot de commande *Commande PIC*.

2.4.4 Étage de réception et étage de stimulation

L'étage analogique de réception responsable de la régulation de l'énergie et du décodage des données est en tout point identique aux circuits utilisés dans les versions antérieures du microstimulateur (MNS et NLB). Le signal RF est redressé à double alternance puis filtré par un condensateur pour ensuite être régulé par une diode zener à seuil ajustable. La démodulation est effectuée par un détecteur d'enveloppe. Enfin, le décodeur Manchester, déjà présenté à la section 1.3, utilise une configuration classique de bascules D et un circuit de délai de $\frac{3}{4}T$.

L'étage de sortie est lui aussi directement inspiré des versions antérieures de l'implant. Un convertisseur numérique à analogique (CNA) génère une tension proportionnelle à l'amplitude de stimulation souhaitée. Cette tension est ensuite convertie en courant par un amplificateur opérationnel configuré en source de courant. Le courant est finalement injecté dans le nerf via une structure de commutateurs en H, permettant d'inverser le sens de circulation du courant dans le nerf.

Les détails du schéma électrique de l'étage d'entrée et de l'étage de sortie sont disponibles sur le schéma du microstimulateur de la Figure B.1.

2.4.5 Contrôleur intégré sur FPGA

La Figure 2.9 présente les principaux blocs constitutifs du FPGA. On y trouve en particulier la file à détection d'entête occupant une bonne partie de l'espace utile du FPGA, où sont reçues et analysées les données transmises par le contrôleur externe. La Figure 2.7 illustre de façon plus détaillée les fonctions de la file à détection d'entête. On voit qu'elle est responsable de la vérification du CRC, de la reconstitution de la trame en éliminant les zéros de remplissage et du décodage du mot de commande.

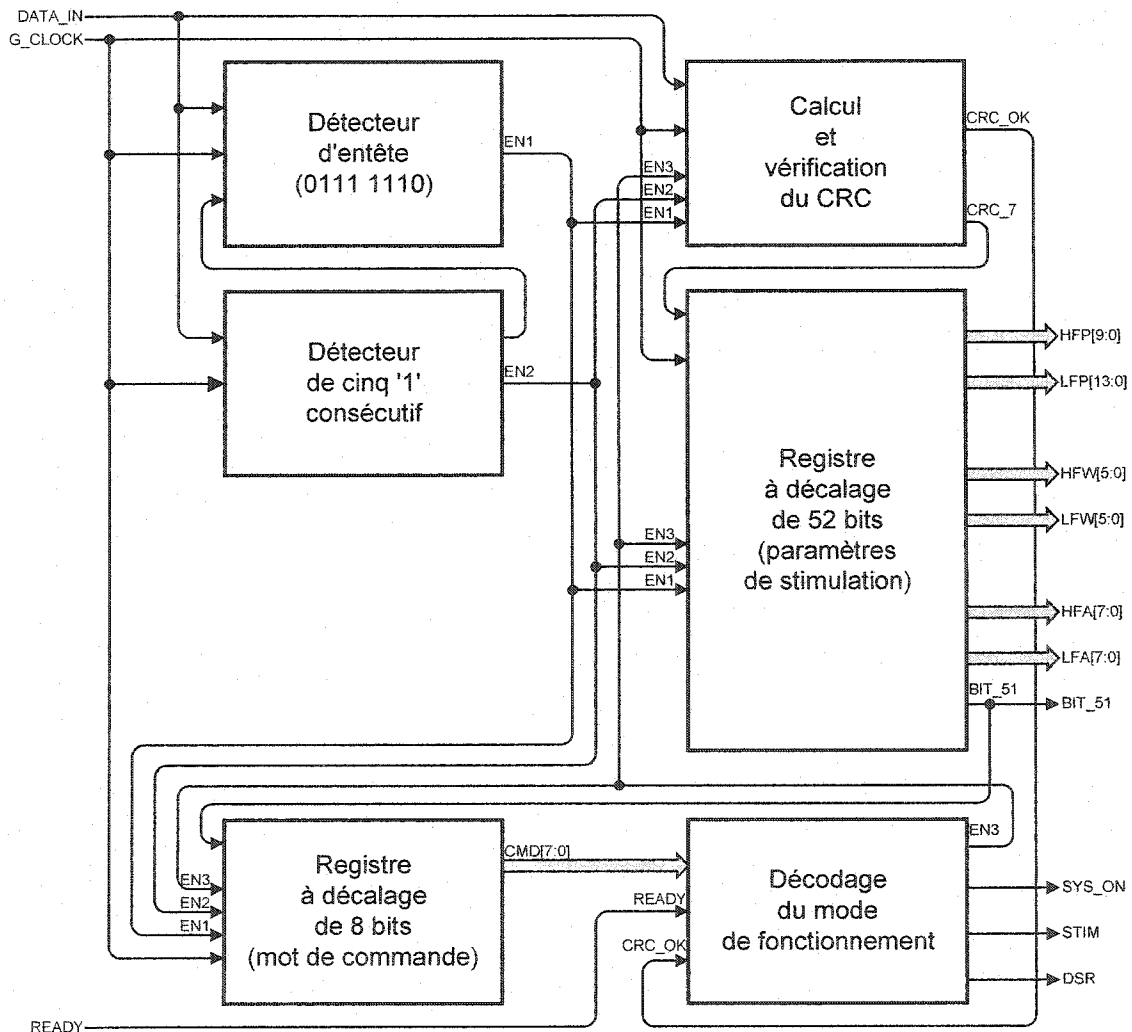


Figure 2.7 : Schéma bloc de la file à détection d'entête du FPGA du NSP

Le fonctionnement de la file à détection d'entête est le suivant : les bits reçus en série sont acheminés simultanément au détecteur d'entête, au détecteur de cinq "1" consécutif, et au vérificateur de CRC. Le vérificateur de CRC, le registre à décalage de 52 bits (paramètres de stimulation) et le registre à décalage de 8 bits (mot de commande) sont cascades en série et peuvent être vus comme un seul et même registre de 68 bits. Ce registre est contrôlé par 3 signaux d'activation *EN1*, *EN2* et *EN3*. Le reste du fonctionnement est présenté sur un diagramme d'état à la Figure 2.8.

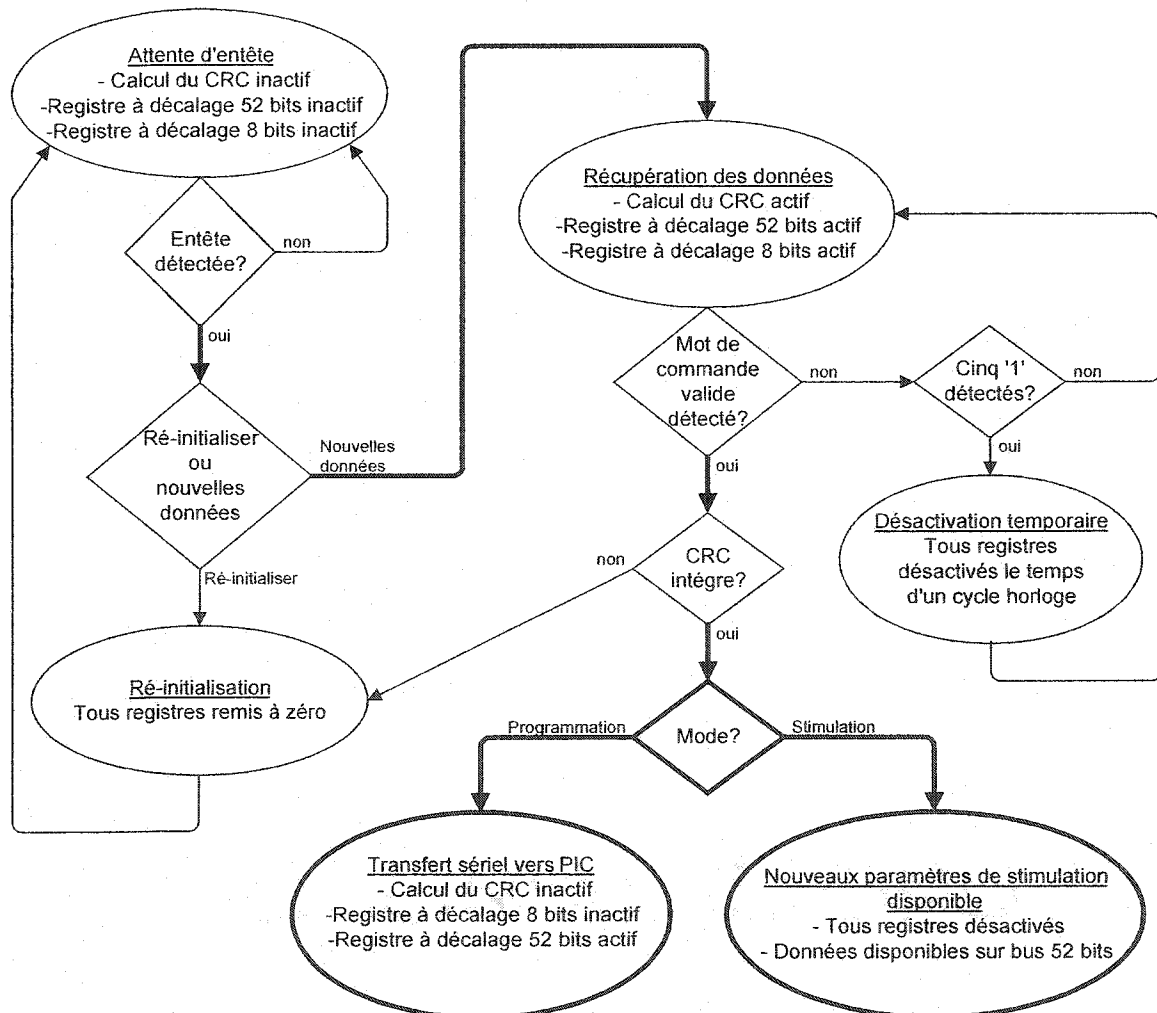


Figure 2.8 : Diagramme d'états de la file à détection d'entête du FPGA du NSP

Les transitions d'états sont contrôlées par l'horloge globale du système de 300 KHz. De plus, il faut noter qu'à tout moment, la file peut être ré-initialisée par une trame de ré-initialisation et lorsque les 68 bits qui suivent l'entête dans une trame valide ont été décalés dans les différents registres correspondants, une commande valide doit nécessairement être détectée, sinon le système se ré-initialisera automatiquement et attendra une nouvelle entête (nouvelle trame).

Une fois les données reçues et disponibles sur le bus interne, le FPGA est donc en mesure de déterminer s'il doit débiter la génération des stimuli ou s'il doit transférer les données au PIC. Dans les 2 cas, il en avise le PIC à l'aide des signaux *SYS_ON* et *DSR* (Figure 2.9). Lorsqu'il reçoit l'accusé de réception du PIC, via le signal *READY*, il peut alors prendre le contrôle (signal *Z_CTRL*) du bus de données externe et débiter une des 2 actions suivantes :

a) Génération de stimuli pour la stimulation sélective

Deux diviseurs de fréquence indépendants et programmables (données *LFP* et *HFP*) génèrent des impulsions à haute et basse fréquence. Une petite machine séquentielle algorithmique (MSA) détecte ces impulsions et supervise alors les différentes étapes de la génération d'un stimulus :

- le convertisseur numérique à analogique (CNA) est programmé (données *LFA* ou *HFA*) et activé (signaux *CS*, *SCLK* et *DIN*)
- les commutateurs de courant sont actionnés (signaux *UP* et *DOWN*) à intervalles réguliers grâce à un système de temporisation programmable (données *LFW* ou *HFw*)
- le CNA est désactivé

- la MSA est prête à traiter la prochaine impulsion. Les impulsions de basse fréquence sont prioritaires sur les impulsions haute fréquence.

b) Transfert des données vers le PIC

L'horloge globale de 300 KHz du système est désactivée. Le sous registre de 52 bits de la file à détection d'entête est activé (Figure 2.7). Son bit le plus significatif (signal *BIT_51*) est connecté au bus de données externe via le signal *DIN*. L'horloge globale du système est contrôlée par le PIC via le signal *SCLK*. Le PIC peut alors recevoir en série les 52 bits de données qui lui sont destinés.

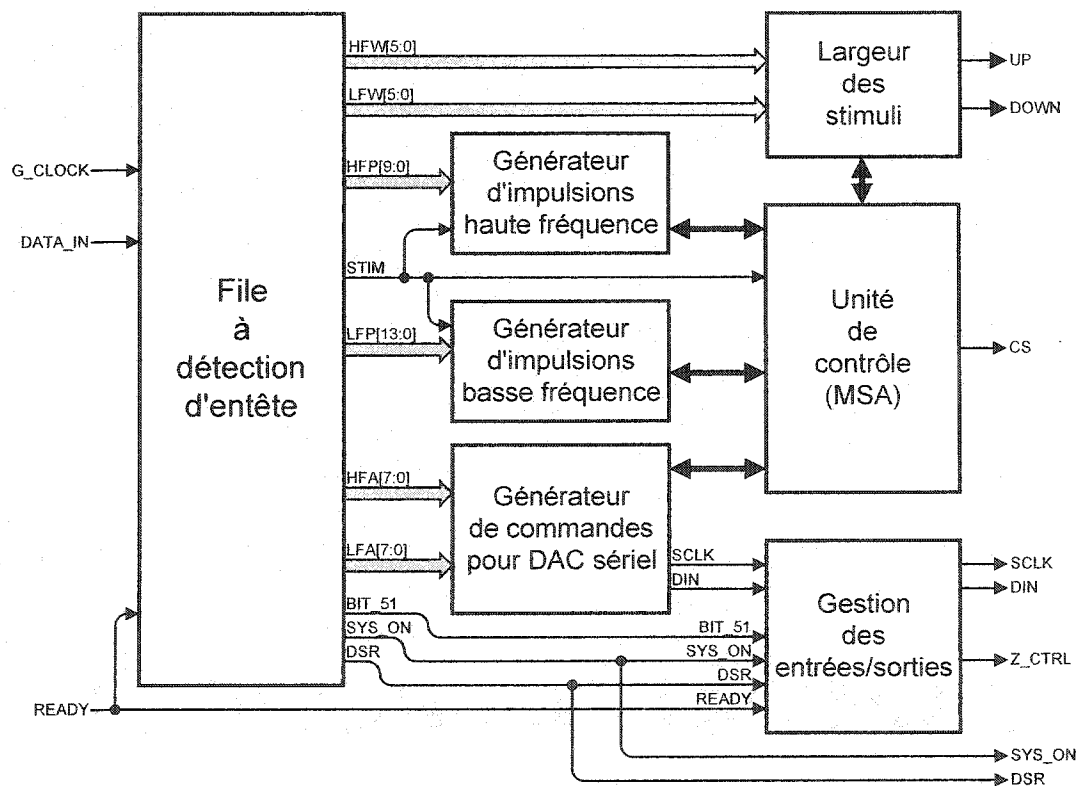


Figure 2.9 : Schéma bloc du contrôleur intégré sur FPGA pour le NSP

2.4.6 Contrôleur intégré sur PIC

Le fonctionnement du PIC est basé sur un programme séquentiel qui est exécuté à une fréquence indépendante de celle du FPGA. Le programme est constitué d'une boucle principale et de 5 sous routines qui définissent les 5 modes de fonctionnement du PIC. Ces derniers correspondent directement aux 5 modes d'opération de l'implant. Seul le mode *Haute impédance* du PIC utilise un nom différent, mais ce dernier correspond au mode *Stimulation sélective* de l'implant. La Figure 2.10 présente la structure de boucle du programme principal du PIC et les détails des transitions de modes.

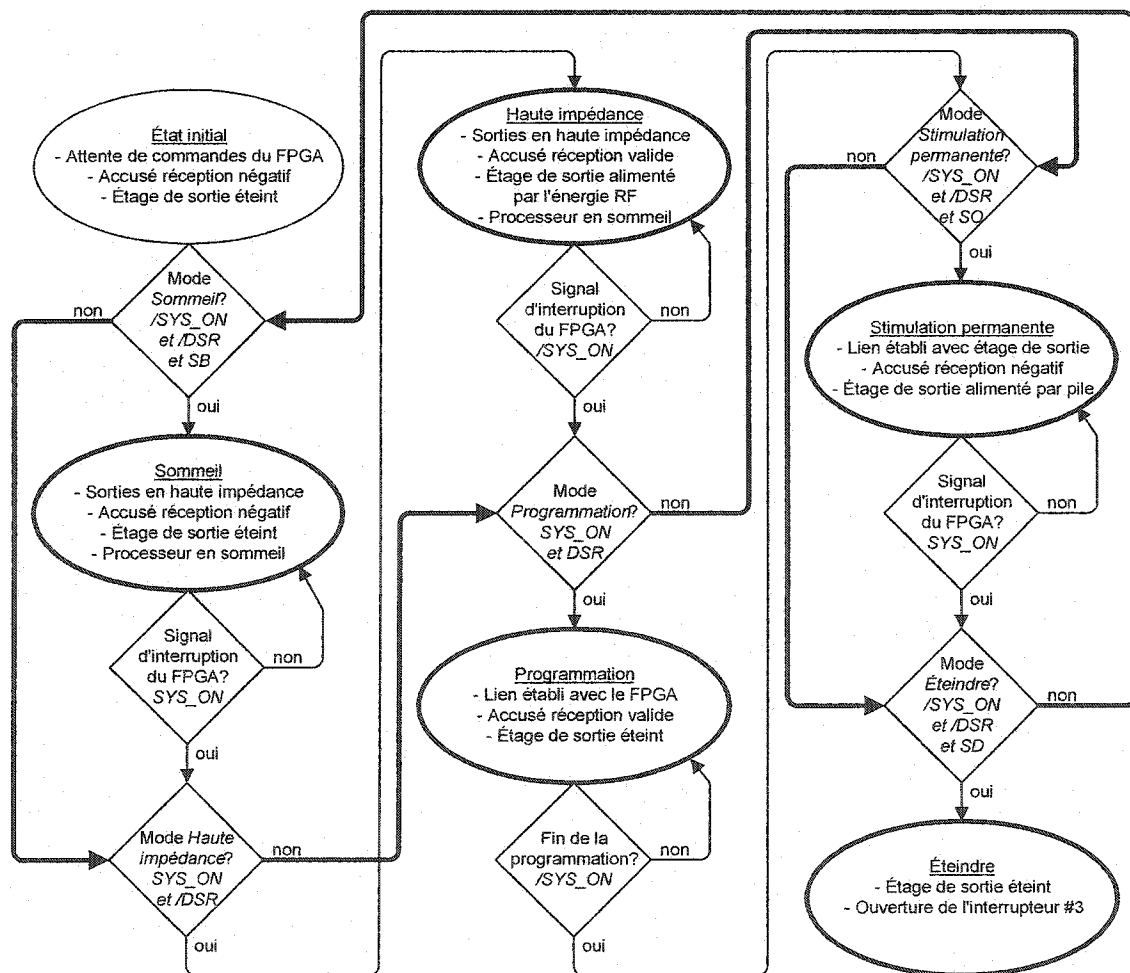


Figure 2.10 : Diagramme d'états du contrôleur intégré sur PIC pour le NSP

Le programme principal détermine le mode d'opération à appeler en fonction des signaux *SYS_ON* et *DSR* et en fonction d'une variable sauvegardée dans un registre interne (*Flags*). Ce registre, qui intervient dans les branchements des sous routines, est directement modifiable en *Mode programmation* par le mot de *Commande PIC* contenu dans la sous trame de données de 52 bits (section 2.4.2). Lorsque le FPGA est désactivé, c'est ce registre qui contient l'information permettant de décider si le PIC doit être désactivé (*Flags[SD]*), se mettre en sommeil (*Flags[SB]*) ou débiter la stimulation permanente (*Flags[SO]*). Le Tableau 2.3 présente le contenu des registres internes du PIC utilisés pour la génération des stimuli et pour la gestion des modes d'opération.

Tableau 2.3 : Les principaux registres du PIC du NSP

Acronyme	Longueur	Sous division	Expression littérale	Valeur min	Valeur max	Résolution ¹
Amp	8 bits	[b7..b0]	$([b7..b0]/255)*I_{\max}$	0 μ A	2 mA	8 μ A
PW	8 bits	[b7..b0]	$[b7..b0]*3*T_{cy}$	0 μ s	2.1 ms	8.4 μ s
Freq	2+2+4 bits	[p1,p0][a1,a0][e3..e0]	$(T_{wd}*2^{[p1,p0]-1} + T_{wd}*[a1,a0] + 510*T_{cy}*[e3..e0])^{-1}$	5.3 Hz	200 Hz	1.2 Hz
Ton	4 bits	[b3..b0]	$(128*[b3..b0])/Freq$	0.6 s	360 s	4 s
Toff	2+2 bits	[r1,r0][p1,p0]	$2*[r1,r0]*T_{wd}*2^{[p1,p0]-4}$	0.4 s	18 s	0.4 s
Flags	4 bits	[SO][ND][SD][SB]	[SO] = Activer la stimulation (Stimulation ON) [ND] = Nouvelles données (New Data) [SD] = Éteindre le PIC (Shut Down) [SB] = Activer le mode sommeil (Stand By)			
I _{max} = Courant maximal de sortie (2mA) T _{cy} = Durée d'un cycle machine (2.8 μ s) T _{wd} = Durée du temporisateur interne (Watch Dog Timer) (24ms) ¹ Résolution autour d'une valeur typique (200 μ A, 175 μ s, 20.3Hz, 18.9s, 12.3s)						

C'est le PIC qui a le contrôle sur les 3 interrupteurs de gestion d'énergie (Figure 2.5). Les 2 premiers, directement commandés par les signaux *BAT_PWR_EN* et *RF_PWR_EN*, lui permettent d'alimenter en énergie l'étage de sortie soit avec la pile, soit avec l'énergie RF. Le troisième interrupteur a une double commande : le PIC peut

seulement désactiver cet interrupteur (signal *SYS_OFF*) alors que le FPGA lui peut seulement l'activer (signal *SYS_ON*). Ainsi il est possible d'activer à distance le PIC (donc l'implant) via le FPGA, mais il est aussi possible de le désactiver en envoyant une commande au PIC qui effectuera automatiquement une interruption de l'alimentation *PIC_PWR* (Figure 2.5) quelques instants après que l'énergie RF ait été retirée, provoquant ainsi une désactivation complète de l'implant. La Figure 2.11 présente le schéma électrique de cet interrupteur à double commandes.

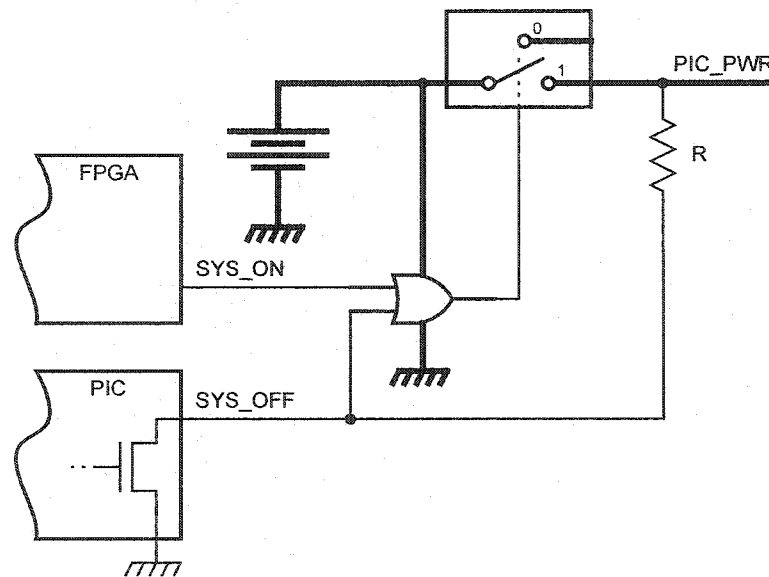


Figure 2.11 : Schéma de l'interrupteur d'alimentation du PIC pour le NSP

Pour le mode de *Stimulation permanente*, la sous routine utilisée consiste, dans l'ordre, à programmer le CNA (signaux *CS*, *SCLK* et *DIN*), activer le signal *UP*, attendre pendant la durée spécifiée par *PW*, commuter *UP* et *DOWN*, attendre de nouveau pendant *PW*, désactiver le CNA, attendre d'une durée $1/Freq$ puis recommencer autant de fois que spécifié par *Ton*. Ensuite, désactiver l'étage de sortie et mettre le PIC en mode *Sommeil*

pendant une durée de T_{off} . Et ainsi de suite jusqu'à ce qu'il y est une interruption matérielle par le signal SYS_ON en provenance du FPGA.

2.5 Réalisation du prototype

Un prototype de ce nouveau microstimulateur a été réalisé sur un circuit imprimé avec des circuits intégrés commerciaux programmables et des composants discrets. Un FPGA 40MX04 d'Actel et un microcontrôleur PIC16F84 de Microchip ont été utilisés. Le CNA sériel de 8 bits MAX550A de Maxim choisi est suffisamment rapide et occupe moins d'espace que le CNA parallèle adopté dans les versions antérieures de l'implant. Les autres circuits intégrés ont été pris dans la famille des circuits CMOS à faible consommation de Texas Instruments. Il ont principalement été choisis pour leur faible encombrement et leur propriétés énergétiques.

Le contrôleur de bus présent sur le schéma bloc détaillé de la Figure 2.5 a été introduit lors de la réalisation du schéma électrique et nécessite l'usage d'un circuit intégré particulier de Texas Instruments, le 74CBTLV3245. Ce dernier permet d'avoir des entrées/sorties en haute impédance lorsque le circuit n'est pas alimenté. Propriété que le FPGA ne possède pas.

La pile utilisée est une pile au Lithium de 500mAh qui devrait permettre de maintenir l'implant opérationnel pendant plus d'un an. En effet, considérons le cas extrême d'une consommation moyenne de $10\mu A$ (I_{low}) du système en période d'inactivité ou de stimulation sélective et de 1 mA (I_{high}) lors de la génération d'un stimulus. Si le système génère des stimuli bipolaire de $300\mu s$ de large (PW) à une fréquence de 30Hz

(Freq) pendant 20s (Ton) et que la période d'inactivité est de 10s (Toff), on obtient à partir de l'équation (1) une période de 0.72s d'activité par minute ($T_{activité}$).

$$T_{activité} = \frac{60}{Ton + Toff} * Ton * Freq * (2 * PW) \quad (1)$$

Donc, l'implant consomme 1mA pendant 1.2% du temps ($T_{\%high}$) et 10 μ A pendant 98.8% du temps ($T_{\%low}$). À partir de l'équation (2), la durée de vie moyenne de la pile sera donc $T_{pile} = 22800$ heures, soit 2 ans ½. Q_{pile} représente la capacité énergétique en mAh de la pile.

$$T_{pile} = \frac{Q_{pile}}{I_{high} * T_{\%high} + I_{low} * T_{\%low}} \quad (2)$$

Bien sûr ce calcul est approximatif et ne tient pas compte des courants transitoires. Mais la marge d'erreur est suffisamment grande pour ne pas avoir à se faire de soucis.

Comme pour le NLB, l'antenne de couplage inductive a été intégrée au PCB par un enroulement concentrique de 3 tours de piste de cuivre suffisamment large pour ne pas dégrader le facteur de qualité.

Pour faciliter l'éventuelle mise à jour du programme dans le PIC après l'assemblage final de l'implant, nous utilisons la fonction de programmation sur circuit du microcontrôleur. Pour cela, nous avons prévu 3 points de branchement sur le microstimulateur pour faciliter la connexion du programmeur sériel.

Le schéma électrique de l'implant ainsi que les schémas logiques du FPGA et le programme assembleur du PIC sont présentés à l'annexe B. Certaines machines à états ont été décrites en VHDL et leurs codes sont aussi présentés en annexe.

La Figure 2.12 présente une photographie à l'échelle 1/1 du microstimulateur assemblé. Certaines modifications ont été apportées par rapport au circuit imprimé initial. Nous verrons la nature de ces modifications ainsi que les performances de ce nouveau microstimulateur au chapitre 4, lors de la présentation des expérimentations animales et les résultats correspondants. Les masques de l'ensemble du circuit imprimé sont présentés à l'annexe B.

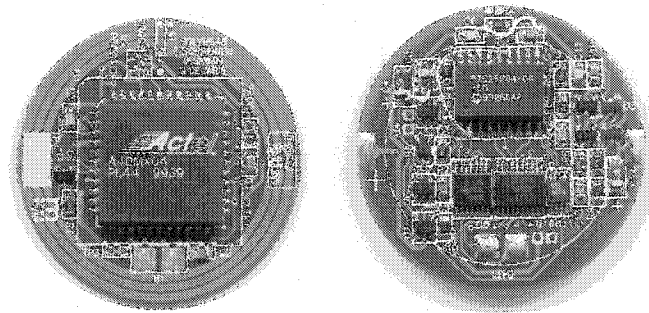


Figure 2.12 : Photographie du NSP à l'échelle 1/1

2.6 Conclusion

À l'issue de ce chapitre, nous pouvons noter que la difficulté, dans la réalisation de ce nouveau microstimulateur, ne résidait pas essentiellement dans l'implantation de la nouvelle fonctionnalité de stimulation permanente. En effet, la nécessité de faire cohabiter les 2 fonctionnalités et de conserver l'ancienne technique d'alimentation par énergie RF a été la principale source de réflexion. Utiliser un seul contrôleur pour gérer l'ensemble des fonctions n'aurait pas pu garantir une stabilité de fonctionnement, surtout au moment de la commutation des alimentations, car des artéfacts sur le signal d'alimentation peuvent corrompre certaines données mémorisées. Comme il n'était pas envisageable de connecter la pile au FPGA, seul la solution des 2 contrôleurs était viable,

ce qui a grandement simplifié le design et accéléré sa réalisation. Il reste maintenant à concevoir un nouveau contrôleur externe capable d'exploiter toutes les nouvelles fonctionnalités de ce microstimulateur. C'est ce que nous vous proposons au chapitre suivant.

CHAPITRE 3

CONTRÔLEURS EXTERNES

3.1 Introduction

Nous avons présenté dans les deux chapitres précédents des microstimulateurs implantables qui nécessitent tous un contrôleur externe pour les alimenter en énergie, mais aussi pour leur transmettre les paramètres de stimulation. Un tel contrôleur est donc constitué d'un amplificateur de puissance inductive RF pour le transfert de l'énergie, d'un modulateur de signal RF pour la transmission des données, d'un encodeur de trame pour la conversion des paramètres en mots binaires et d'une interface utilisateur pour la sélection des paramètres ou des modes de fonctionnement. De plus, dans le cas du NLB, un module supplémentaire est nécessaire pour décoder puis afficher l'impédance mesurée par le microstimulateur sous-cutané. La Figure 3.1 présente le schéma bloc générique d'un contrôleur externe complet.

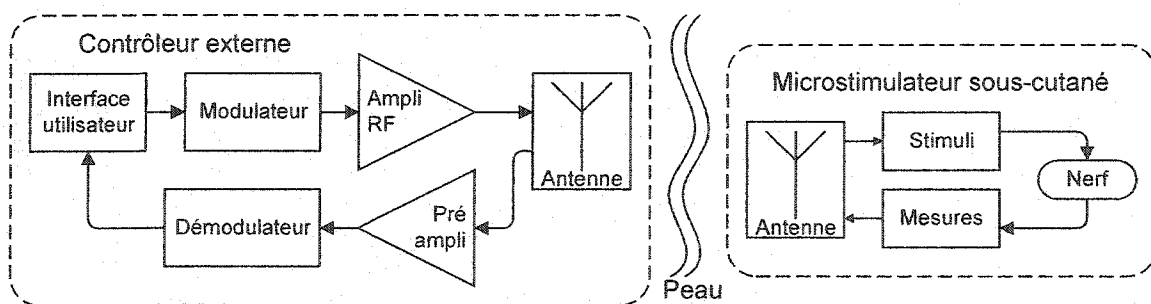


Figure 3.1 : Schéma bloc générique d'un contrôleur externe

En pratique, on peut distinguer deux sortes de contrôleurs externes : les contrôleurs de test flexibles (CTF) et les contrôleurs portatifs simplifiés (CPS).

- Les CTF utilisent un ordinateur de type PC comme interface utilisateur et permettent d'effectuer les premiers tests sur les implants ou encore de mettre au point les séries de paramètres qui seront utilisées dans les CPS. Ces CTF ont l'avantage d'être rapide et simple à développer et permettent d'utiliser toutes les fonctionnalités des implants.

- Les CPS sont conçus pour être utilisés par les patients (ou les techniciens animaliers dans notre cas) et ne disposent que de fonctions limitées et d'une interface utilisateur simplifiée. Ces CPS sont généralement basés sur des microcontrôleurs, des petits écrans à cristaux liquides (LCD) et trois ou quatre touches de fonction.

Par la suite, une variante du contrôleur portatif a été développée pour permettre aux médecins de disposer des fonctionnalités d'un CTF avec l'encombrement d'un CPS : le contrôleur portatif flexible (CPF). Ce chapitre présente les contrôleurs externes qui ont dû être développés pour permettre d'exploiter les nouvelles fonctionnalités des microstimulateurs.

3.2 Couplage magnétique et transfert de puissance

Quelque soit le type de microstimulateur ou de contrôleur externe, le même circuit de transfert de puissance inductive est utilisé. Le principe de transfert par couplage inductif et la circuiterie de l'étage de sortie RF des contrôleurs sont présentés dans cette section.

La Figure 3.2 présente la configuration de base sur laquelle repose le transfert de puissance par couplage inductif de la plupart des systèmes électroniques implantables ne disposant pas de source d'énergie interne [18], [39]. Les éléments du circuit résonnant

primaire (L_p , C_p et R_p) font partie de l'étage de sortie du contrôleur externe, tandis que les éléments du circuit résonnant secondaire (L_s , C_s et R_s) font partie de l'étage de réception du microstimulateur sous-cutané. Dans la suite des développements, on supposera que les résistances R_p et R_s incluent aussi, respectivement, la résistance de sortie de l'amplificateur et la résistance équivalente du microstimulateur.

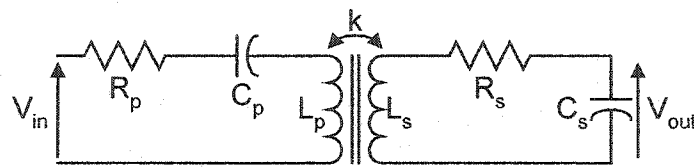


Figure 3.2 : Topologie d'un circuit à couplage inductif

Lorsque les deux circuits résonnants sont réglés à la même fréquence f_0 , le circuit couplé résultant possède lui aussi une résonance à la fréquence f_0 . La Figure 3.3 présente la variation de la tension induite V_{out} dans le secondaire, en fonction de la fréquence f pour différentes valeurs du coefficient de couplage k entre les 2 bobines.

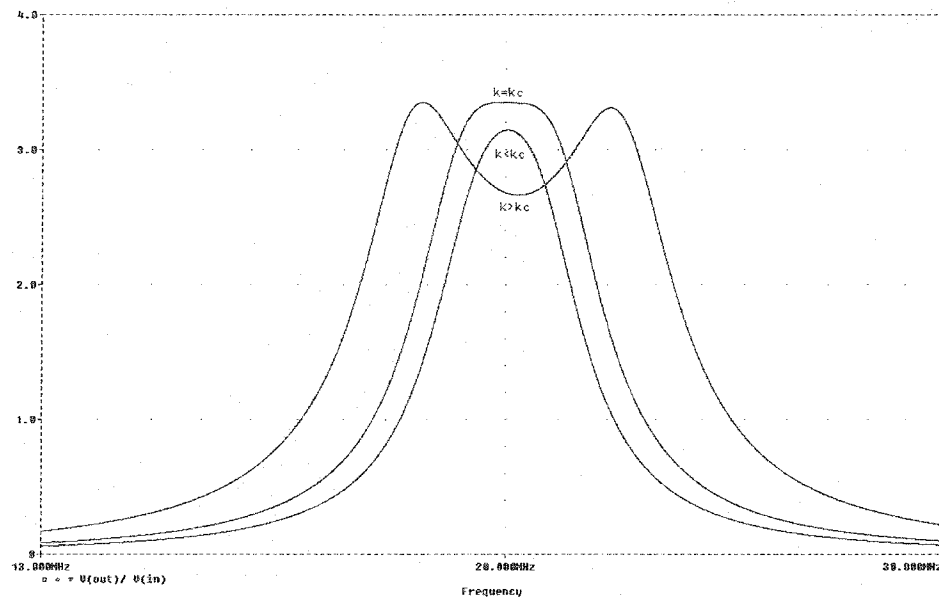


Figure 3.3 : Variation du rapport V_{out}/V_{in} pour différentes valeurs de k

À la résonance, la tension induite V_{out} du secondaire est reliée à la tension d'excitation V_{in} du primaire par la relation (1). Q_p et Q_s représentent respectivement les facteurs de qualité des circuits primaire et secondaire, avec $Q_p = \omega.L_p/R_p$ et $Q_s = \omega.L_s/R_s$.

$$\frac{V_{out}}{V_{in}} = \frac{k\sqrt{L_s/L_p}}{k^2 + 1/(Q_p.Q_s)} \quad (1)$$

Le taux de transfert maximal est obtenu lorsque le coefficient de couplage k est égal au coefficient de couplage critique k_c qui est donné par la relation (2).

$$k_c = \frac{1}{\sqrt{Q_p.Q_s}} \quad (2)$$

Ainsi, la tension induite, donc l'énergie transmise, sera maximale à la fréquence f_0 lorsque le primaire et le secondaire seront réglés à la même fréquence de résonance f_0 et lorsque le couplage, donc la distance, entre les deux bobines atteindra sa valeur critique k_c . Le rapport de tension maximal V_{out}/V_{in} correspondant est obtenu en combinant les équations (1) et (2)

$$\left(\frac{V_{out}}{V_{in}}\right)_{MAX} = \frac{1}{2}\sqrt{Q_p.Q_s}\sqrt{\frac{L_s}{L_p}} \quad (3)$$

À la vue de ces résultats, on constate qu'il est important de respecter les 4 critères suivants pour garantir le bon fonctionnement du lien inductif :

- S'assurer d'une bonne correspondance entre les deux fréquences de résonance des circuits primaire et secondaire.
- Contrôler la précision et la stabilité de la fréquence du signal d'excitation du primaire autour de la résonance.

- Ajuster la distance de couplage critique en modifiant les facteurs de qualité Q_p et Q_s soit en variant L_p et L_s soit en réglant R_p .
- Contrôler la puissance transmise en ajustant l'amplitude du signal d'excitation du primaire.

L'amplificateur RF de puissance choisi pour exciter le circuit primaire au niveau des contrôleurs externes, est un amplificateur de classe D, dont la topologie est présentée à la Figure 3.4. Un tel amplificateur nécessite généralement une paire de transistors (T_1 et T_2) en commutation et un circuit résonant de reconstruction. Dans notre cas, ce circuit résonant sera aussi le circuit primaire du lien inductif. Les transistors T_1 et T_2 agissent comme 2 commutateurs, qui permettent de stimuler le circuit résonant avec une onde carrée dont la fréquence de commutation coïncide avec la fréquence de résonance, permettant ainsi d'entretenir les oscillations. Un tel amplificateur permet d'atteindre des rendements supérieurs à 90%, en supposant que les pertes dans les transistors ainsi que les temps de commutation sont suffisamment faibles.

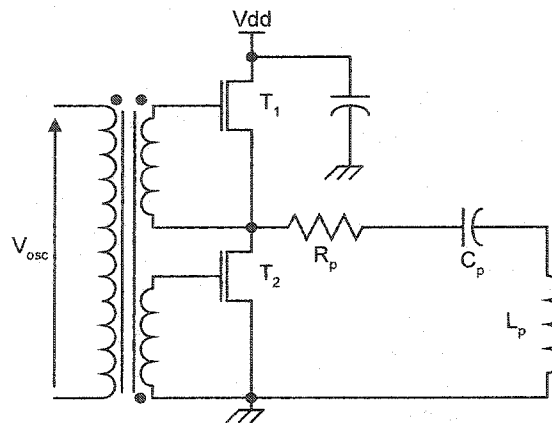


Figure 3.4 : Topologie d'un amplificateur de classe D

Le schéma électrique complet de l'étage de sortie RF des contrôleurs, incluant le modulateur, l'amplificateur de puissance et le circuit primaire résonant, est présenté à la Figure 3.5. Le modulateur ASK utilise un amplificateur sélectif à transistor (Q1) en rétroaction positive.

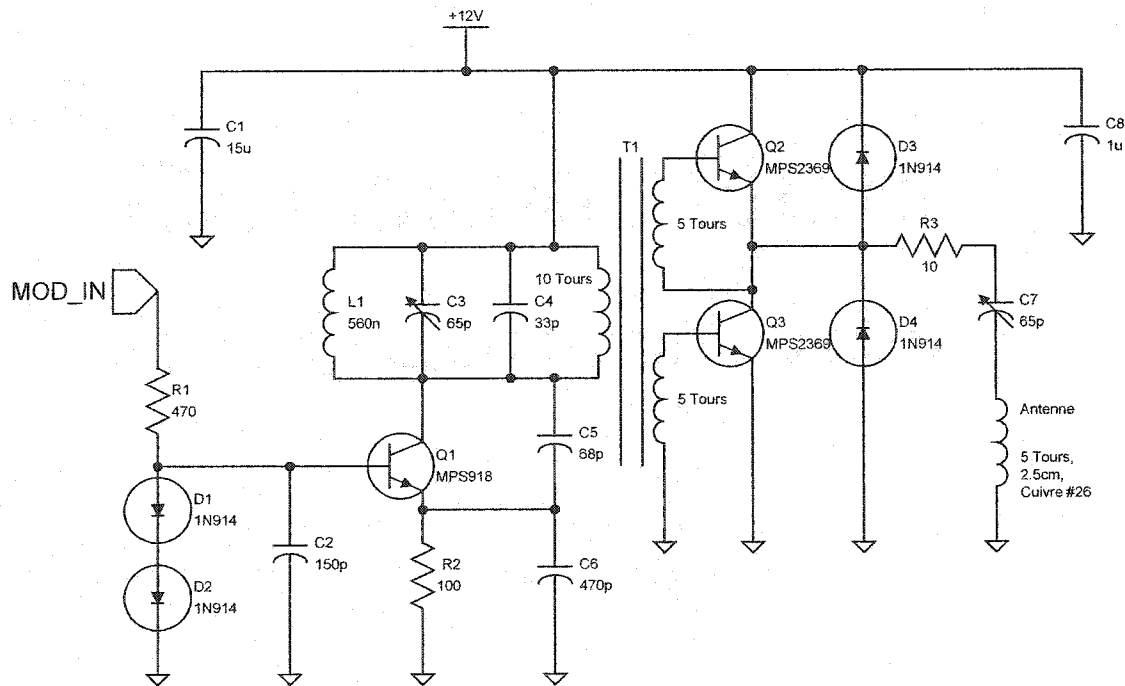


Figure 3.5 : Schéma électrique de l'émetteur ASK des contrôleurs externes

3.3 Contrôleurs de test flexibles

Chaque nouveau microstimulateur se différencie de son prédécesseur par un nouveau protocole de communication et par de nouvelles fonctionnalités. Le mode de transfert de l'énergie et le type de modulation quand à eux restent inchangés. Il est donc possible de réaliser un contrôleur externe générique, qui permettrait de commander l'ensemble des nouveaux implants en sélectionnant simplement l'encodeur de trame spécifique à chaque microstimulateur. Pour permettre cette flexibilité, le CTF utilise un

ordinateur de type PC et un module sur port parallèle comme interface vers le modulateur ASK. Un logiciel permet de choisir le type d'implant et permet de saisir au clavier les paramètres de stimulations souhaités pour chacun des modes.

La Figure 3.6 présente la structure de l'interface parallèle du CTF, qui garantit le taux de transfert de 600Kbps vers le modulateur, donc vers l'implant. En effet, il est difficile de maintenir un taux de transfert constant d'un port parallèle d'ordinateur, ou de n'importe quel autre port du PC, principalement à cause des interruptions matérielles qui peuvent survenir à tout moment.

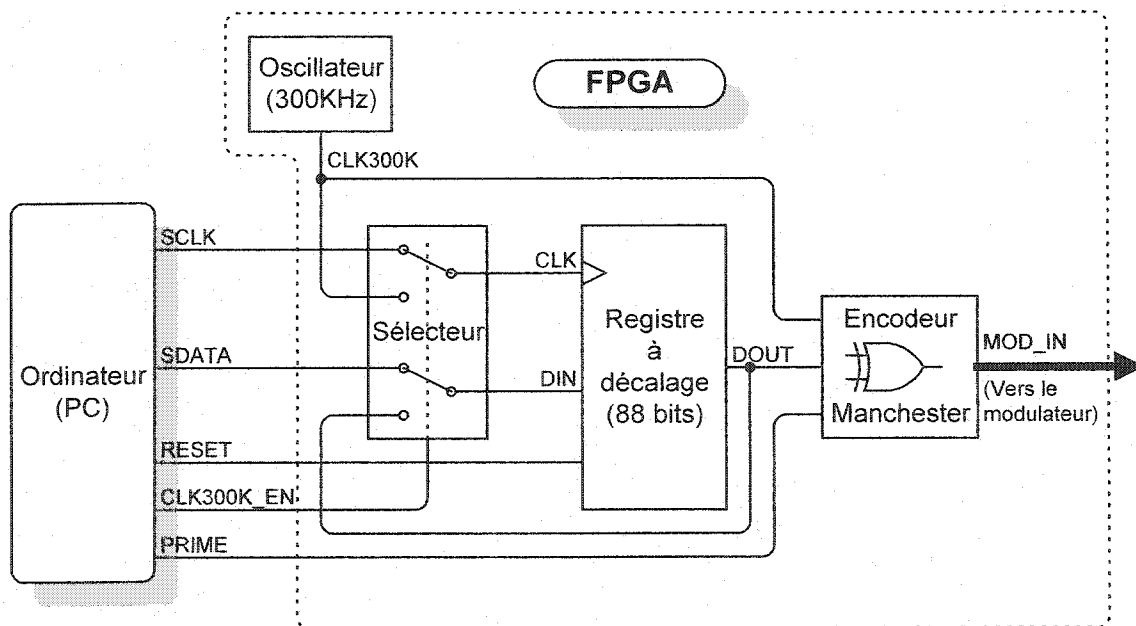


Figure 3.6 : Schéma bloc de l'interface PC-modulateur du CTF

La trame de données formée dans l'ordinateur est transférée dans un registre à décalage, faisant partie d'un circuit FPGA, à une vitesse libre fixée par le signal *SCLK* du PC. Une fois les données transférées, l'horloge du registre est commutée sur un oscillateur à 300KHz de rapport cyclique 50%. Le codage au format Manchester est

assuré par un OU-exclusif entre l'horloge à 300KHz et les données sauvegardées dans le registre à décalage. Un signal *PRIME* commandé par l'ordinateur, agit sur l'encodeur Manchester pour en forcer la sortie à "1" et permet d'activer le modulateur ASK un peu avant l'envoi effectif des données.

La Figure 3.7 présente un exemple d'écran de configuration du CTF pour une stimulation permanente. Les schémas électriques et logiques du CTF, ainsi que le code source du logiciel, sont présentés à l'annexe C.

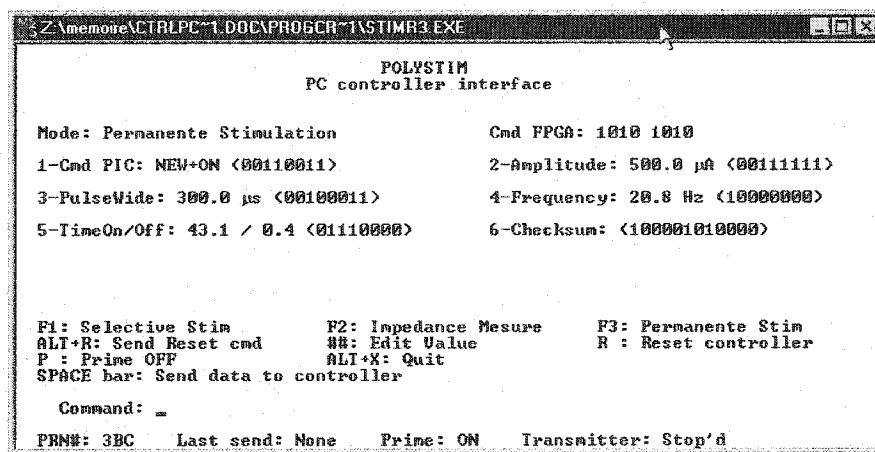


Figure 3.7 : Exemple d'écran de l'interface utilisateur du CTF

Le CTF est utilisé sur une base quotidienne pour le test des implants nouvellement assemblés ou pour le test d'endurance d'un implant NSP témoins qui est soumis à une simulation d'utilisation intensive dans une solution saline à +40°C.

3.4 Contrôleurs portatifs

Les contrôleurs portatifs utilisent eux aussi le même étage de sortie RF que le CTF, mais leur interface utilisateur a été largement simplifiée. Si le CTF permet de couvrir toute la plage des paramètres dans chacun des modes, le CPS lui ne permet

d'effectuer que 8 formes de stimulation par mode de fonctionnement (6 paramètres pour chaque stimulation). Ces ensembles de paramètres sont sélectionnés par le chirurgien en fonction des besoins du sujet (chien). La Figure 3.8.a présente la structure du CPS avec son microcontrôleur central (PIC) qui gère l'écran LCD (Liquid Crystal Display) et le codage des trames. Deux touches de fonction permettent de sélectionner le numéro de la stimulation souhaitée, puis la touche *SEND* permet d'activer le transfert d'énergie et d'information vers le microstimulateur sous-cutané.

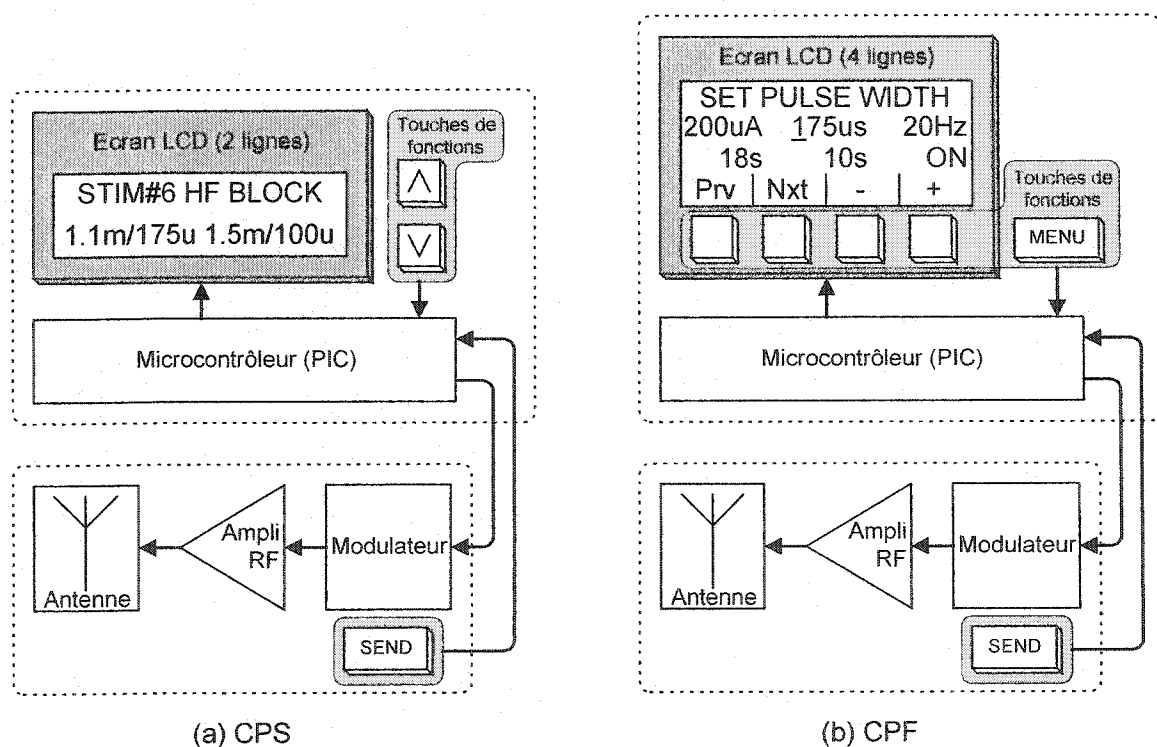


Figure 3.8 : Schémas blocs des contrôleurs portatifs simplifié (a) et flexible (b)

Pour augmenter la flexibilité du contrôleur portatif, le CPF de la Figure 3.8.b a été développé sur la base du CPS avec un écran LCD plus grand (4 lignes) et des touches de fonctions à menu contextuel. La touche *MENU* permet de choisir le type d'implant et son mode de fonctionnement, tandis que les 4 touches de fonction situées à proximité de

l'écran LCD permettent de faire la sélection d'un paramètre et de le modifier progressivement. La fonction de chacune de ces touches est spécifiée sur la dernière ligne du LCD et dépend du menu courant.

Le schéma électrique du CPS ainsi que le code source assembleur du microcontrôleur utilisé (PIC) sont présentés à l'annexe C.

Le CPS est utilisé quotidiennement depuis plus d'un an par les techniciens chargés du suivi des animaux d'expérimentation. Il n'a nécessité à ce jour que des ajustements du circuit de couplage inductif et des réparations mécaniques dues à l'usure des boutons poussoirs.

3.5 *Module d'affichage d'impédance*

Pour éviter de perturber le circuit résonnant primaire du lien inductif au niveau du contrôleur externe, le système dédié à la récupération des données envoyées par le NLB, est un module indépendant qui dispose de sa propre antenne et vient se placer entre le contrôleur externe et le patient. De plus, le protocole utilisé par le NLB étant un protocole standard de transmission sériel asynchrone, il peut directement être branché sur n'importe quel système qui possède un UART (Universal Asynchronous Receiver Transmitter). De tels UART étant disponibles sur un ordinateur et sur un PIC, la connexion du module aux contrôleurs se trouve simplifiée.

La Figure 3.9 présente l'architecture choisie pour démoduler le signal FSK provenant du NLB. Le circuit intégré LM3089 de National Semiconductor est spécialisé dans la démodulation des signaux FM dans la bande intermédiaire IF de 10.7MHz et utilise un circuit détecteur en quadrature. Ce circuit a l'avantage d'intégrer dans un même

boîtier, en plus du détecteur en quadrature permettant de la démodulation, un pré-amplificateur, un détecteur du niveau de la porteuse et un amplificateur de sortie. Ainsi, nous avons simplement adapté le pré-amplificateur d'entrée à notre antenne inductive à 2MHz, en nous assurant de filtrer le signal à 20MHz provenant du contrôleur. Le signal démodulé (sortie du filtre passe bas) et sa composante continue (tension d'offset à la sortie du moyennneur) sont traités par un comparateur à hystérésis qui donne les niveaux logiques compatibles. Le détecteur de porteuse nous permet de synchroniser l'UART du contrôleur (PC ou PIC) avec le début de la transmission des données du NLB. En effet, tant que le signal *CD* (Carrier detected) est inactif (aucune porteuse détectée) le contrôleur connecté au module de réception ignore les éventuels artefacts sur le signal *DATA_ASYNC*, puis lorsqu'une porteuse est détectée ($CD = "1"$) l'UART du contrôleur est activé et attend le bit de départ de la trame asynchrone.

Si ce module de réception est utilisé avec le PIC, la sortie du comparateur peut être branchée directement à l'entrée de l'UART, mais s'il est utilisé avec un PC, il est nécessaire de rajouter un convertisseur de niveau logique pour respecter le standard RS232.

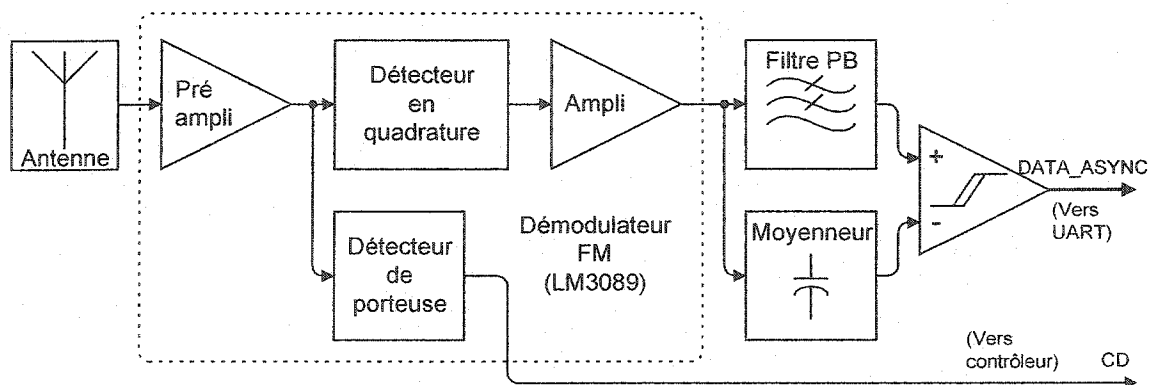


Figure 3.9 : Schéma bloc du module de réception de données FSK

Pour pouvoir correctement interpréter le signal transmis par le NLB, l'UART de réception devra être configuré pour recevoir des mots de 8 bits avec une parité paire, un bit de départ et 2 bits d'arrêt à une vitesse de 4800 bps. Puisque les données sont envoyées en boucle par le NLB, le récepteur pourra accumuler plusieurs échantillons et les comparer avant de les valider. De plus, il est important de noter que ce mot de 8 bits n'est qu'une évaluation de la fréquence d'oscillation du VCO dédié à la mesure de l'impédance nerf-électrodes. Il incombe donc au contrôleur externe de calculer l'impédance équivalente à partir du temps T_{eval} et du coefficient de proportionnalité du VCO (voir chapitre 1). Des exemples de mesures sont présentés au chapitre 4.

Le schéma électrique complet du prototype de ce module de réception est présenté à l'annexe D. Ce prototype n'a été testé que sur le CTF lors de la validation en laboratoire du NLB. Les contrôleurs portatifs n'ont pour le moment pas encore été adaptés pour ce module.

3.6 Stabilisation de la porteuse par contrôle numérique

Nous avons vu à la section 3.2 qu'il est important de contrôler la fréquence de la porteuse pour garantir un transfert maximal de puissance. Le modulateur ASK choisi pour les contrôleurs externes actuels utilise un oscillateur analogique à transistors, dont la fréquence est ajustable par un condensateur variable. Il est à noter qu'un tel oscillateur est simple et compact, mais il n'est pas possible de garantir la stabilité et la précision de la fréquence en fonction de la température, ou encore suite à de simples chocs mécaniques sur le boîtier du contrôleur. Une adaptation de l'interface parallèle du CTF a été développée autour d'un PLL (Phase Lock Loop) numérique pour permettre un contrôle

numérique précis de la fréquence de la porteuse. De plus, afin de minimiser la phase de recouvrement lors de la commutation des transistors de l'amplificateur de classe D de sortie, nous avons inclu un générateur d'impulsions à deux phases sans recouvrement. Un tel système permet d'éliminer le transformateur de la Figure 3.4 en commandant directement les transistors T_1 et T_2 par deux signaux numériques sans recouvrement.

La Figure 3.10 présente l'architecture détaillée de l'interface parallèle modifiée du CTF. On y reconnaît toute la partie destinée au transfert des données de trame déjà présentée à la section 3.3 : le sélecteur, le registre à décalage (ici 96 bits plutôt que 88 bits) et l'encodeur Manchester. Par contre, la sortie du codeur Manchester n'est pas directement envoyée au modulateur analogique (Figure 3.5), mais plutôt à un générateur d'impulsions sans recouvrement où sera faite la modulation (porte ET sur la Figure 3.10).

Les autres blocs de la Figure 3.10 servent à asservir la fréquence de la porteuse. Le PLL externe au FPGA est constitué d'un détecteur de phase regroupant un "Charge-Pump", un filtre RC passe bas et un VCO dont la sortie compatible TTL est utilisée par le générateur d'impulsions sans recouvrement du FPGA pour contrôler les transistors de l'amplificateur de classe D.

Le détecteur de phase compare la phase du signal de référence issu du diviseur par 20 (F_{ref}) et la phase du signal issue du diviseur par N (F_{div}). Lorsque F_{div} est en retard (respectivement en avance) par rapport à F_{ref} , le condensateur C est déchargé (chargé) à travers la résistance R. La tension résultante aux bornes du condensateur C (V_C) sert de tension de contrôle pour le VCO. Lorsque la tension V_C diminue (augmente) la fréquence

d'oscillation du VCO diminue aussi (augmente), réduisant ainsi le déphasage entre F_{ref} et F_{div} jusqu'à ce que la boucle se stabilise et que F_{div} soit égale à F_{ref} (déphasage nul).

L'oscillateur de référence utilise un cristal de quartz de très grande précision à 1MHz qui permet de générer à la sortie du diviseur par 20 une fréquence de référence précise $F_{ref} = 50\text{KHz}$. Le diviseur de fréquence par N est quant à lui programmable à travers un registre de 12 bits contrôlé par l'ordinateur. La fréquence générée par le VCO, donc la fréquence de la porteuse, est donnée par : $F_{porteuse} = N \cdot F_{ref}$. Ce qui donne une résolution théorique de 50KHz. En pratique la stabilité de la fréquence dépend aussi de la stabilité du VCO et de la stabilité de la tension V_C aux bornes du condensateur.

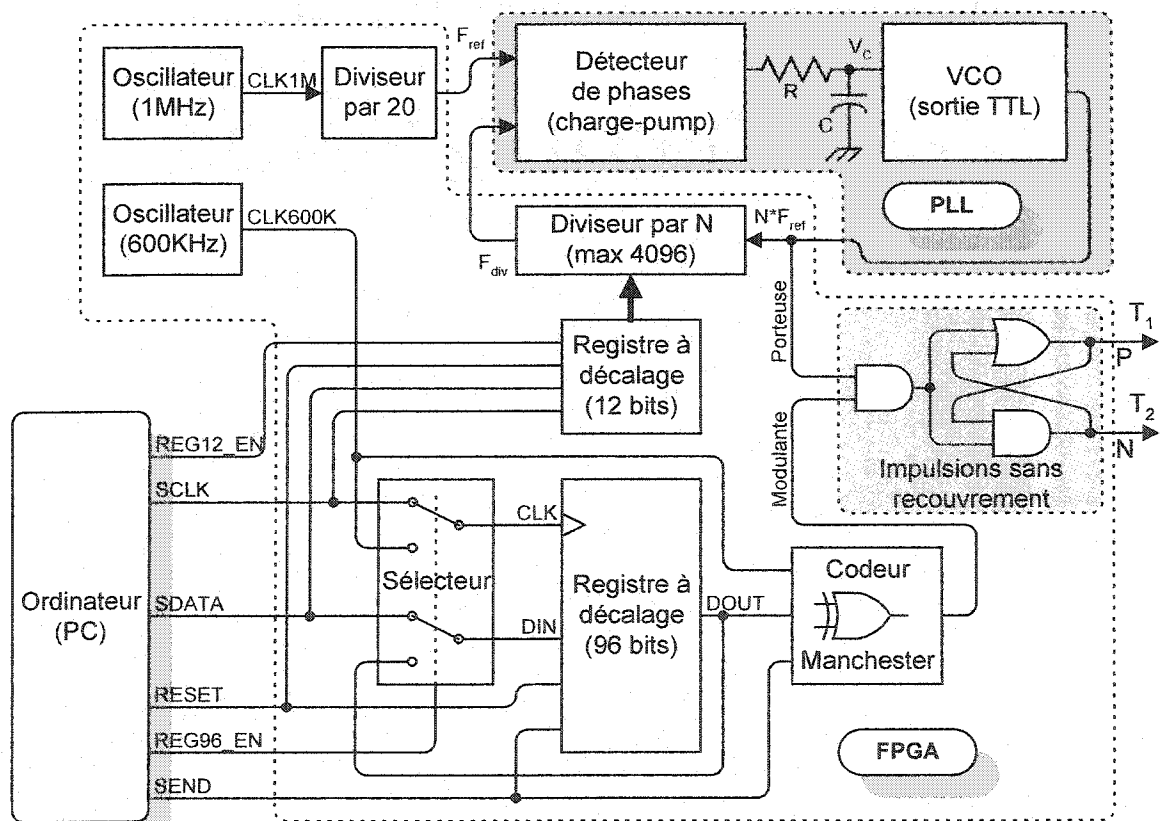


Figure 3.10 : Schéma bloc détaillé d'un CTF à contrôle numérique de la porteuse

Un prototype de cette interface a été réalisé avec un PLL 74HC4046 de Texas Instruments et un FPGA 40MX02 d'Actel sur un circuit imprimé. Ce prototype a permis de valider le principe de fonctionnement et a démontré l'utilité de réglage du nouveau modulateur, ainsi qu'une amélioration de 15% de l'énergie transférée au microstimulateur. Cependant les tests n'ont pas pu être réalisés à haute fréquence et à pleine amplitude, car le contrôle direct des transistors de l'amplificateur de classe D nécessite le remplacement du transistor T_1 de type n de la Figure 3.4 par un transistor de type p, limitant ainsi la fréquence de commutation. De plus la tension d'alimentation de l'amplificateur ne peut pas dépasser la tension d'alimentation du FPGA. Les tests ont donc été conduits à 10MHz et 5V sur un implant modifié pour les circonstances. L'étape de contrôle de l'amplificateur nécessitera encore quelques améliorations, mais le principe d'asservissement de fréquence est incontestablement un gain majeur pour le transfert de puissance vers le microstimulateur sous-cutané.

3.7 Conclusion

Nous avons présenté succinctement dans ce chapitre l'évolution des différents contrôleurs externes. Les détails de la conception de l'interface utilisateur qui constituent une grande partie du travail n'ont toutefois pas été rapportés. Heureusement les outils de programmation et les microcontrôleurs d'aujourd'hui simplifient l'étape de réalisation. Mais la recherche de convivialité des systèmes demande beaucoup d'attention et nécessite de nombreux réajustements avant d'obtenir un produit adapté aux besoins d'un patient non initié aux subtilités de l'électronique. Enfin, le deuxième point délicat des contrôleurs externes reste encore l'émetteur RF qui est essentiel pour une bonne

alimentation en énergie des implants et une fiabilité du lien de communication. Le contrôle numérique de la fréquence de la porteuse permettra d'améliorer le rendement du lien, mais le couplage entre les deux bobines n'étant pas constant il sera nécessaire de rajouter un dispositif d'asservissement de la puissance transmise ou encore une régulation du coefficient de couplage par le contrôle de R_p . Les détails de ces contraintes sont abordés au prochain chapitre, où nous présentons les résultats de nos expérimentations animales (chiens) à long terme des implants et de leurs contrôleurs.

CHAPITRE 4

EXPÉRIMENTATIONS ET RÉSULTATS

4.1 Introduction

Les chapitres précédents de ce mémoire ont été réservés à la présentation de nos travaux de conception et de réalisation des microstimulateurs implantables et de leurs contrôleurs externes. L'étape suivante, de grande importance, consiste en l'évaluation de ces systèmes dans un contexte réel d'utilisation lors d'expérimentations animales en phase chronique. De plus, chacun de ces systèmes de stimulation neurale fait l'objet d'une validation *in vitro* avant de telles expérimentations. Ce chapitre présente les résultats des tests des deux microstimulateurs NLB et NSP du point de vue de leurs fonctionnalités et de leurs performances. Par la suite nous présentons les précautions à prendre et les contraintes à respecter afin de pouvoir utiliser de tels systèmes *in vivo* chez l'animal. En effet, l'utilisation de dispositifs électroniques dans un milieu biologique nécessite une isolation électrique des circuits face au milieu hostile dans lequel ils se trouvent. De plus, une bio-compatibilité du dispositif implanté face aux tissus qui l'entourent est obligatoire pour éviter tout phénomène de rejet ou d'infection. Nous présentons aussi dans ce chapitre le protocole expérimental appliqué aux 6 sujets (chiens) qui ont fait l'objet de l'étude et nous analysons les différents résultats obtenus.

4.2 Tests fonctionnels et analyse des performances du NLB

Les résultats présentés dans cette section proviennent de tests in vitro des prototypes à PCB réalisés. Il est à noter que pour chacun des systèmes, un premier prototype réalisé sur plaquette expérimentale a permis de mettre au point la majeure partie du design, éliminant ainsi les risques d'erreurs de conception sur le circuit imprimé final. Ainsi, les deux microstimulateurs (NLB et NSP) ont pu être opérationnels dès leur première mise sous tension, mais quelques modifications ont tout de même dû être apportées, car certaines parties n'avaient pu être complètement testées sur les plaquettes expérimentales.

La fonction de stimulation sélective du NLB a immédiatement été fonctionnelle, mais la mesure d'impédance a nécessité de nombreux réajustements du circuit d'adaptation d'impédance entre l'amplificateur du signal FSK de sortie et l'antenne inductive. Dans un premier temps, l'adaptation d'impédance permettait un très bon transfert du signal FSK vers le récepteur externe, mais dégradait beaucoup trop le circuit de réception de l'énergie, rendant l'implant très peu efficace pour la stimulation sélective (courant maximal de stimulation inférieur à 1mA au lieu de 2mA). C'est ainsi que nous avons rajouté dans le circuit d'adaptation d'impédance, une résistance (R_L dans le schéma de la Figure 1.6(b)) pour réduire l'efficacité du circuit FSK mais améliorer la réception d'énergie. Compte tenu de la proximité des antennes, le signal FSK transmis demeure facilement détectable par le contrôleur externe.

4.2.1 Caractérisation de la stimulation sélective

Une des spécifications du NLB est d'être entièrement compatible avec le MNS, il doit donc pouvoir générer le même type de stimuli sélectifs avec la même précision et les mêmes plages. Le Tableau 4.1 regroupe les mesures comparatives des 6 paramètres de stimulation sélective pour le MNS et le NLB. On constate que les plages, résolutions et précisions y sont parfaitement comparables comme on pouvait s'y attendre.

Tableau 4.1 : Comparaison des performances du MNS et du NLB pour la stimulation sélective

Paramètre	Basse fréquence			Haute fréquence		
	Amplitude	Fréquence	Largeur d'impulsion	Amplitude	Fréquence	Largeur d'impulsion
Plage MNS	0 – 1.83 mA	18.5 – 2380 Hz	0 – 212 μ s	0 – 1.83 mA	301 – 2380 Hz	0 – 212 μ s
Plage NLB	0 – 1.90 mA	18.6 – 2150 Hz	0 – 215 μ s	0 – 1.90 mA	298 – 2150 Hz	0 – 215 μ s
Résolution ¹ MNS	10 μ A	< 70 mHz	4 μ s	10 μ A	1 Hz	4 μ s
Résolution ¹ NLB	10 μ A	< 70 mHz	3.9 μ s	10 μ A	1 Hz	3.9 μ s
Précision ² MNS	7.7 %	0.4 %	0.6 %	7.7 %	0.2 %	3.1 %
Précision ² NLB	7.5 %	0.4 %	0.9 %	7.4 %	0.2 %	2.9 %

¹ Résolution autour d'une valeur typique (0.9mA, 30Hz, 175 μ s; 1.3mA, 600Hz, 100 μ s).
² Erreur par rapport à la valeur souhaitée pour une valeur typique.

La Figure 4.1 présente le temps de montée de la source d'énergie inductive pour un couplage critique entre le microstimulateur et le contrôleur externe ainsi qu'un exemple de stimuli sélectifs générés par le NLB. Le bruit qu'on remarque sur l'alimentation DC et sur le signal de stimulation vient de la porteuse RF, qui ne peut pas être entièrement filtrée par le circuit de régulation et qui induit aussi des courants non négligeables dans les sondes de l'oscilloscope de mesure. D'autres caractéristiques importantes comme celles-ci sont résumés dans le Tableau 4.2, afin de définir certaines

limites d'utilisation du NLB en ce qui a trait à sa rapidité ou à ses performances énergétiques.

Tableau 4.2 : Autres caractéristiques importantes du NLB

Temps de réponse à une commande de stimulation	5.3 ms
Temps de montée de l'alimentation DC	3.5 ms
Taux de rejection de la porteuse	-39 dB
Puissance reçue par le microstimulateur	30 mW*
Distance de couplage critique (à l'air libre)	3.5 cm
Taux de montée de l'étage de stimulation	0.3 V/ μ s
* Pour un couplage critique à l'air libre	

Le temps de réponse global du système à une commande de stimulation sélective est tout à fait négligeable et imperceptible par l'utilisateur, mais il dépendra beaucoup de la qualité du couplage entre le microstimulateur et le contrôleur externe. Tant que le niveau d'alimentation DC ne sera pas stable, le décodage des données ne pourra pas être garanti. De plus, la puissance réelle reçue par le microstimulateur implanté est égale à 35% de la valeur nominale disponible dans un couplage critique (optimal) à l'air libre.

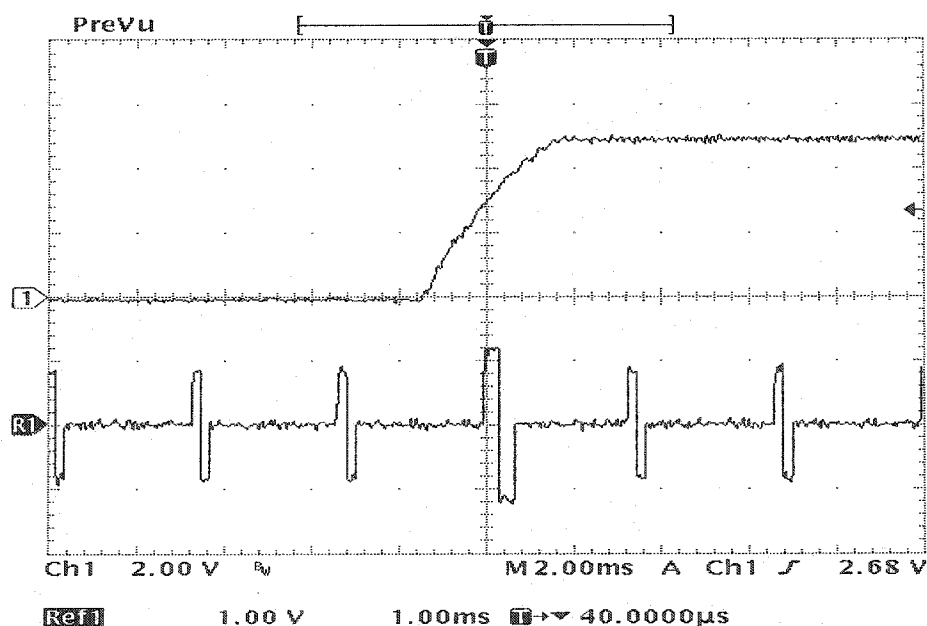


Figure 4.1 : Temps de montée de l'alimentation (Ch1) et signal de stimulation sélective (Ref1) pour le NLB

4.2.2 Caractérisation de la mesure d'impédance

La seconde fonctionnalité du NLB est d'évaluer la composante résistive de l'interface nerf-électrodes et de retourner cette mesure à un afficheur externe. Il est donc nécessaire de caractériser le VCO qui est utilisé pour la conversion tension-fréquence, car le coefficient de proportionnalité intervient directement dans le calcul effectif de la résistance.

La courbe de réponse réelle du VCO de la Figure 4.2(a) révèle une non linéarité entre la tension d'entrée et la fréquence du signal de sortie. Toutefois, cette courbe peut être linéarisée par régression linéaire dans quatre intervalles de fréquences distincts (Figure 4.2(b)). Ainsi, en fonction de la fréquence mesurée, il suffira de choisir l'équation linéaire la plus adaptée et en déduire la tension correspondante. Le Tableau 4.3 présente, en fonctions des intervalles de fréquences, les paramètres de ces 4 équations linéaires mises sous la forme de l'équation (1). Cette équation exprime V_{stim} en fonction de F_{osc} , car c'est la conversion fréquence à tension qui nous intéresse pour ce VCO.

$$V_{stim} = k_{VCOi} * F_{osc} + V_{0i} \quad (1)$$

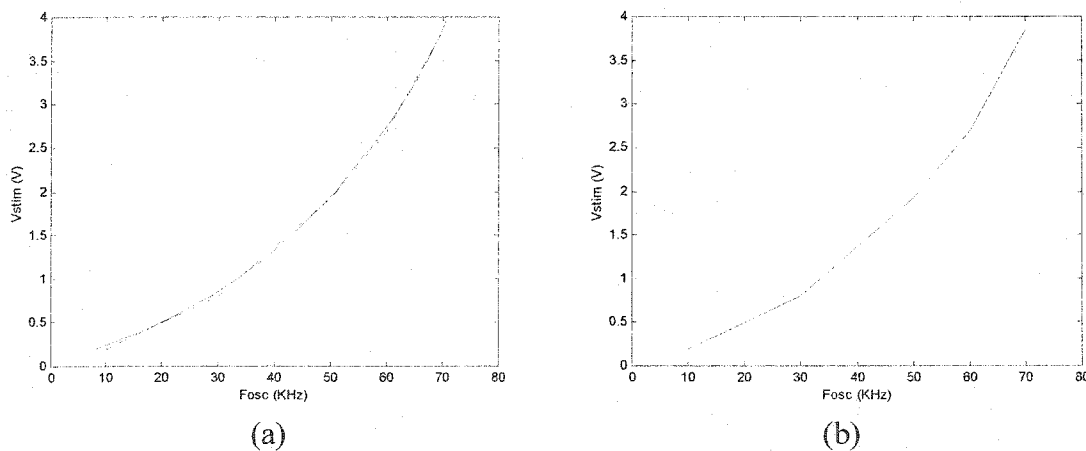


Figure 4.2 : Courbe de réponse réelle (a) et modélisée (b) du VCO

Tableau 4.3 : Coefficients de proportionnalité du VCO en fonction des intervalles de fréquences

Numéro d'intervalle (i)	Intervalles de fréquences	k_{VCOi}^* (V/KHz)	V_{0i} (V)
1	10 – 30 KHz	$3.29 \cdot 10^{-2}$	-0.14
2	30 – 50 KHz	$5.62 \cdot 10^{-2}$	-0.89
3	50 – 60 KHz	$8.09 \cdot 10^{-2}$	-2.12
4	60 – 70 KHz	$11.59 \cdot 10^{-2}$	-4.27
* Valeurs obtenues par régression linéaire sur une dizaine d'échantillons par intervalle de fréquences			

A partir du modèle de la courbe de réponse caractéristique du VCO, nous pouvons maintenant évaluer n'importe quelle tension résultante à la sortie de la source de courant en autant que celle-ci reste dans la plage normale de fonctionnement. Si une fréquence située en dehors de la plage normale de fonctionnement était détectée ($F_{osc} < 10\text{KHz}$), il suffirait simplement d'ajuster le courant de stimulation en conséquence. La valeur reçue par l'afficheur externe représente le nombre de périodes (N_{osc}) contenues dans le signal issue du VCO pendant une durée T_{eval} . Ainsi la fréquence de ce signal est donnée par l'équation (2) et la tension correspondante à l'entrée du VCO (donc la tension à la sortie V_{stim} de la source de courant du stimulateur) est déterminée par l'équation (3). La constante k_{VCOi} est le coefficient de proportionnalité du VCO dans l'intervalle de fonctionnement linéaire "i".

$$F_{osc} = N_{osc} / T_{eval} \quad (2)$$

$$V_{stim} = k_{VCOi} * (N_{osc} / T_{eval}) + V_{0i} \quad (3)$$

Cependant, même si on néglige l'erreur de conversion du VCO, le principe de mesure de la fréquence introduit une erreur inversement proportionnelle à N_{osc} . En effet, la durée T_{eval} est plus exactement liée à la période du signal issue du VCO (T_{osc}) par la relation (4), car le compteur qui évalue la fréquence du VCO compte les périodes entières

et non les fractions de période. Ainsi l'erreur relative sur la mesure de la fréquence (δ_ϵ) est donnée par la relation (5), ce qui se traduit par l'erreur relative maximale (δ_1) de l'équation (6).

$$T_{\text{eval}} = (N_{\text{osc}} + \epsilon) * T_{\text{osc}} \quad \text{tel que } 0 < \epsilon < 1 \quad (4)$$

$$\delta_\epsilon = \epsilon / (N_{\text{osc}} + \epsilon) \quad (5)$$

$$\delta_1 = 1 / (N_{\text{osc}} + 1) \quad (6)$$

On constate alors que l'erreur sera d'autant plus faible que la valeur reçue par l'afficheur externe sera grande. Compte tenu de la flexibilité du système dans la programmation des paramètres T_{eval} et I_{stim} , cette proportionnalité de l'erreur relative peut être vue comme une précision variable pour la mesure de l'impédance. En effet, si une première mesure donne une valeur N_{osc} reçue trop faible par rapport à la précision souhaitée, il suffira d'augmenter la durée T_{eval} ou dans certains cas le courant de stimulation I_{stim} pour obtenir un N_{osc} plus grand et donc une meilleure précision.

Cette caractéristique du système de mesure d'impédance s'est avérée être en pratique un inconvénient quant à la simplicité d'utilisation du dispositif, qui nécessite trop de manipulations et oblige à une bonne maîtrise de l'architecture interne du microstimulateur. Il serait donc nécessaire de développer au niveau du contrôleur de l'afficheur externe un algorithme de convergence itératif permettant de mesurer automatiquement l'impédance du nerf sans que l'utilisateur n'ait à effectuer lui même plusieurs mesures.

Un exemple de signaux intervenant dans la mesure d'impédance est présenté à la Figure 4.3. On peut y voir le signal d'oscillation issu du VCO proportionnel à la tension

maximale aux bornes du nerf et la tension résultante à une des bornes du nerf. Le bruit observé sur les signaux vient des courants induits dans les sondes de l'oscilloscope par le champ magnétique à 20MHz de la porteuse. Pour cet exemple, la fréquence du signal de stimulation est relativement élevée pour permettre de visualiser sur le même écran les 2 signaux, mais en pratique le courant de stimulation sera généralement choisi dans une bande de fréquence autour de 500Hz. Les autres paramètres pour la mesure étaient les suivant :

- courant de stimulation $I_{stim} = 200\mu A$
- résistance témoin mise à la place du nerf (à mesurer) $R_{nerf} = 3.3K\Omega$
- temps d'évaluation $T_{eval} = 7.5ms$
- valeur reçue par le module externe $N_{osc} = 205$

A partir de ces données et des équations (2) et (3), on peut calculer la fréquence d'oscillation $F_{osc} = 27.3KHz$ et la tension correspondante $V_{stim} = 0.76V$ en utilisant les coefficients de l'intervalle de fréquences "1" du Tableau 4.3. Ces valeurs calculées sont tout à fait comparables aux valeurs directement mesurées sur l'implant à l'aide de l'oscilloscope.

Finalement, pour calculer la résistance du nerf, il suffit de résoudre l'équation (7) où R_i représente la résistance de contrôle dans la source de courant (Figure 1.5). Ainsi $R_{nerf}(\text{calculée}) = 3.33K\Omega$ à partir de l'équation (8) et avec $R_i = 470\Omega$.

$$V_{stim} = (R_{nerf} + R_i) * I_{stim} \quad (7)$$

$$R_{nerf} = (V_{stim} / I_{stim}) - R_i \quad (8)$$

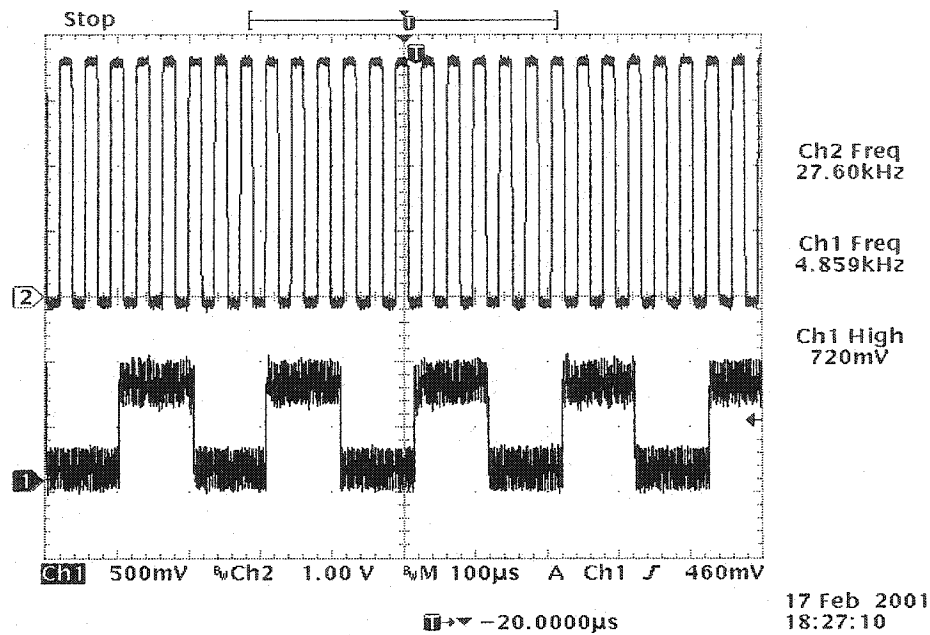


Figure 4.3 : Exemple de signaux intervenants dans la mesure d'impédance du NLB : oscillation du VCO (Ch2) et signal de stimulation (Ch1)

4.3 Tests fonctionnels et analyse des performances du NSP

Tout comme pour les autres microstimulateurs, le NSP a lui aussi fait l'objet d'une mise au point sur plaquette expérimentale avant sa réalisation sur circuit imprimé (PCB). Cependant, toute la partie de la gestion d'énergie n'a pu être testée avant l'assemblage du PCB pour la simple raison que les circuits montés en surface (les seuls disponibles) sont difficilement manipulables sans les circuits imprimés adaptés. Ainsi, ces mises au point ont finalement dû être faites sur le PCB lui même.

Lors de sa première mise sous tension, le NSP semblait pouvoir générer les formes d'ondes pour chacune des stimulations sélective et permanente, mais la commutation entre ces deux modes de stimulation n'était pas convenable. Le problème était relié au FPGA qui se trouvait être alimenté indirectement par des "courants de fuite"

en l'absence de l'énergie inductive, ce qui empêchait la ré-initialisation des signaux de synchronisation entre le PIC et le FPGA (signaux *SYS_ON* et *DSR* sur la Figure B.1) et perturbait le fonctionnement prévu du PIC.

Sur le schéma électrique original du microstimulateur de la Figure B.1 on remarque que les régulateurs U3 et U4 ont leur sorties en commun, permettant ainsi de choisir la source d'énergie qui doit alimenter le circuit de stimulation. Cependant, de tels régulateurs possèdent une diode de protection contre les courants inverses qui connecte la sortie à l'entrée. Le régulateur U3 alimentait alors indirectement le circuit du FPGA lorsque le régulateur U4 était activé. Sur le schéma électrique corrigé présenté à la Figure B.2 le régulateur U3 a simplement été remplacé par un transistor bipolaire de type P. On perd ainsi la régulation (sans que cela affecte le fonctionnement du circuit) mais on assure au moins une bonne isolation des deux circuits d'alimentation en énergie.

La seconde "fuite de courant" provenait du signal *READY* en logique positive qui permet au PIC d'acquiescer au FPGA l'usage du bus externe. C'est certainement l'architecture des tampons d'entrée-sortie du FPGA qui permettait à ce dernier d'être partiellement alimenté en énergie, via une broche d'entrée tout à fait ordinaire, lorsqu'un niveau logique haut y était appliqué. Une simple résistance de "pull-up" et une sortie à drain ouvert ont permis d'isoler parfaitement les deux sources d'énergie. Le schéma électrique global corrigé ainsi que les modifications sur le dessin des masques sont présentés à l'Annexe B.

4.3.1 Caractérisation de la stimulation sélective

Comme pour toutes les autres versions des microstimulateurs, le NSP doit être en mesure d'effectuer la stimulation sélective. Le Tableau 4.4 dresse un bilan comparatif entre le MNS et le NSP afin de vérifier leur entière compatibilité.

La Figure 4.4 présente le temps de montée de la source d'énergie inductive pour un couplage critique entre le microstimulateur et le contrôleur externe ainsi qu'un exemple de stimuli sélectifs générés par le NSP. On constate une moins bonne réjection de la porteuse que dans le NLB (Figure 4.1), sans que cela n'affecte l'étage de stimulation dont le rapport signal sur bruit de 42dB reste tout à fait acceptable.

Tableau 4.4 : Tableau comparatif des performances du MNS et du NSP pour la stimulation sélective

Paramètre	Basse fréquence			Haute fréquence		
	Amplitude	Fréquence	Largeur d'impulsion	Amplitude	Fréquence	Largeur d'impulsion
Plage MNS	0 – 1.83 mA	18.5 – 2380 Hz	0 – 212 μ s	0 – 1.83 mA	301 – 2380 Hz	0 – 212 μ s
Plage NSP	0 – 1.95 mA	18.3 – 1970 Hz	0 – 210 μ s	0 – 1.95 mA	295 – 1970 Hz	0 – 210 μ s
Résolution ¹ MNS	10 μ A	< 70 mHz	4 μ s	10 μ A	1 Hz	4 μ s
Résolution ¹ NSP	9 μ A	< 80 mHz	3.7 μ s	9 μ A	1 Hz	3.7 μ s
Précision ² MNS	7.7 %	0.4 %	0.6 %	7.7 %	0.2 %	3.1 %
Précision ² NSP	7.7 %	0.6 %	1.1 %	6.2 %	0.6 %	3.0 %

¹ Résolution autour d'une valeur typique (0.9mA, 30Hz, 175 μ s; 1.3mA, 600Hz, 100 μ s).
² Erreur par rapport à la valeur souhaitée pour une valeur typique.

Le Tableau 4.5 regroupe d'autres caractéristiques importantes du NSP. On remarquera en particulier que le temps de réponse global du système est relativement long (comparativement au NLB) car la synchronisation du PIC et du FPGA du microstimulateur utilise quelques temporisateurs afin de filtrer les transitions et les

artefacts qui peuvent survenir lors de la phase initiale transitoire de mise sous tension.

Néanmoins, ce délai restera toujours imperceptible du point de vue de l'utilisateur.

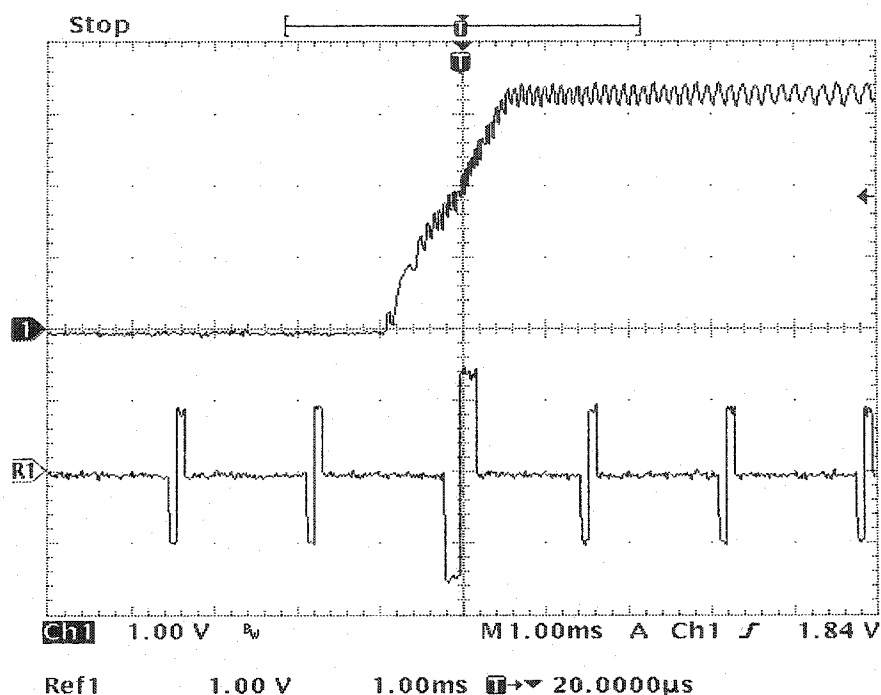


Figure 4.4 : Temps de montée de l'alimentation (Ch1) et signal de stimulation sélective (Ref1) pour le NSP

Tableau 4.5 : Autres caractéristiques importantes du NSP

Temps de réponse à une commande de stimulation sélective	49 ms
Temps de réponse à une commande de stimulation permanente	153 ms
Temps de montée de l'alimentation DC	1.5 ms
Taux de réjection de la porteuse	-35 dB
Puissance reçue par le microstimulateur (à l'air libre)	28 mW*
Distance de couplage critique (à l'air libre)	3.0 cm
Taux de montée de l'étage de stimulation	0.1 V/ μ s
* Pour un couplage critique à l'air libre	

Le taux de montée de 0.1V/ μ s pour l'étage de stimulation de sortie semble être insuffisant pour notre application. Cependant, cet effet restera négligeable devant les délais introduits par la capacité parasite du nerf qui se situe dans les dizaines de nano-Farades.

4.3.2 Caractérisation de la stimulation permanente

La caractérisation de la stimulation permanente consiste d'une part à la vérification de la précision des paramètres réels de stimulation par rapport aux paramètres programmés, mais elle implique aussi de vérifier la fiabilité de l'implant lors de la commutation des modes d'opération. Ceci permet de garantir la bonne gestion de l'énergie pour respecter une longue durée de vie de la pile. Le Tableau 4.6 présente les performances du NSP dans son mode de stimulation permanente, mais la fiabilité et la durée de vie réelle de la pile du microstimulateur ne peuvent être vérifiées qu'à long terme lors des tests in vivo. Ces résultats seront donc présentés à la section 4.6.

Tableau 4.6 : Performances du NSP lors de la stimulation permanente

Paramètre	Amplitude	Fréquence	Largeur D'impulsion	Durée Allumé	Durée Éteint
Plage	0 – 0.95 mA	5.2 – 360 Hz	0 – 2120 μ s	0.6 – 365 s	0.5 – 20 s
Résolution ¹	9 μ A	1 Hz	8.7 μ s	4 s	< 1 s
Précision ²	9 %	1 %	4.6 %	1 %	2 %
¹ Résolution autour d'une valeur typique (200 μ A, 20.3Hz, 175 μ s, 18.9s, 12.3s).					
² Erreur par rapport à la valeur souhaitée pour une valeur typique.					

Les plages et résolutions de la fréquence de stimulation présentées dans le Tableau 4.6 peuvent être obtenues en utilisant toutes les ressources du système, toutefois il est recommandé d'utiliser certaines fréquences particulières pour minimiser la consommation d'énergie. En effet, le contrôle de la période du signal de stimulation est en partie assuré par un temporisateur interne au PIC (le Watch Dog Timer) qui continue à fonctionner même lorsque l'horloge globale du microcontrôleur est interrompue. La durée maximale de ce temporisateur étant de 24ms, si on choisit une fréquence de stimulation telle que sa période soit un multiple de 24ms, le système ne sera activé que

pour générer un stimulus et restera dans son mode *sommeil* entre chaque impulsion. Les 7 fréquences recommandées sont : 41.6Hz, 20.8Hz, 13.8Hz, 10.4Hz, 8.3Hz, 6.9Hz et 5.9Hz.

La Figure 4.5 illustre un signal de stimulation permanente généré par le NSP ainsi que le signal d'horloge du PIC qui n'est activé que quelques centaines de micro-secondes avant la formation d'un stimulus puis est désactivé jusqu'à la prochaine interruption. On notera cependant que la fréquence réelle de l'horloge (1MHz) est largement supérieure à ce qu'elle paraît être sur l'exemple présenté. En effet, la période d'affichage choisie, pour visualiser le signal de stimulation de basse fréquence, réduit la vitesse d'échantillonnage de l'oscilloscope numérique (critère de Nyquist non respecté) qui ne permet alors pas d'afficher fidèlement le signal à 1MHz.

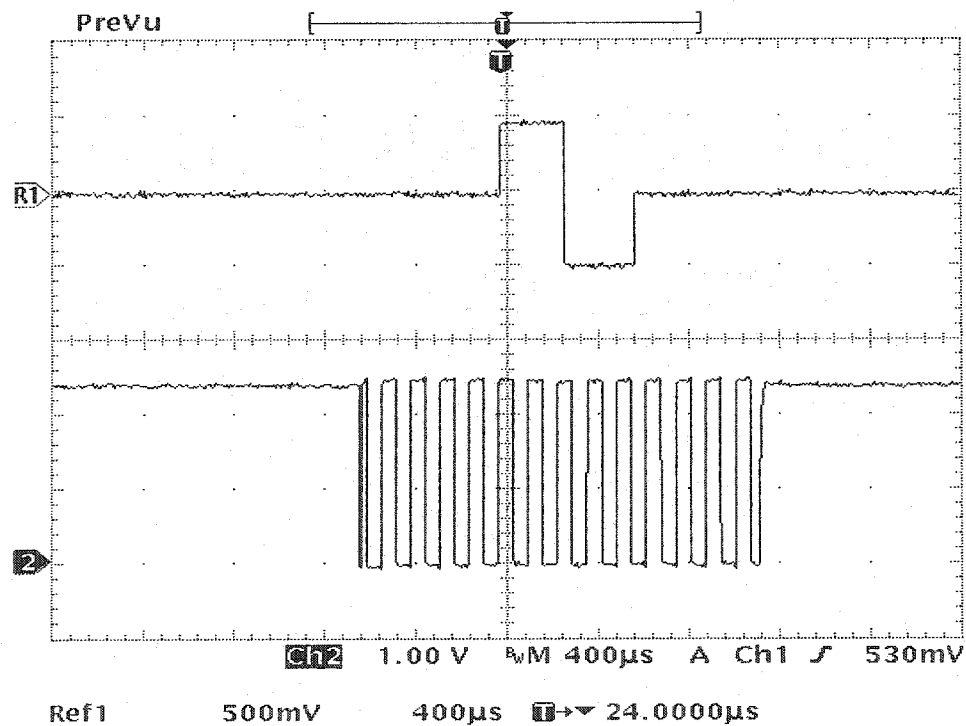


Figure 4.5 : Signal de stimulation permanente (Ref1) et horloge du PIC(Ch2) pour le NSP

Bien que les tests fonctionnels préliminaires semblent tous être positifs, il reste à vérifier l'opération du microstimulateur dans un milieu réel contraignant et hostile pour les implants électroniques. C'est ce que nous abordons dans les prochaines sections.

4.4 Préparation du matériel pour les tests en phase chronique

4.4.1 Assemblage des microstimulateurs

L'utilisation des microstimulateurs pour des tests en phase chronique chez l'animal implique de nouvelles contraintes d'ingénierie qui ne sont pas nécessairement reliées au domaine de l'électronique. Prenons par exemple l'assemblage des 8 exemplaires du NSP. Bien qu'on ne parle pas ici de production de masse, il a tout de même fallu mettre au point une procédure d'assemblage et de test pour pouvoir garantir la même fiabilité de fonctionnement pour tous les exemplaires. De plus, pour ne pas avoir à faire des réglages fins sur chacun des implants, les composants ont été choisis et agencés de sorte que leurs imprécisions ne viennent pas dégrader les performances d'un microstimulateur par rapport à un autre. Parfois, des systèmes trop bien optimisés sont difficilement reproductibles. Ainsi, chaque implant est partiellement assemblé par des techniciens avant une première phase de tests de toutes les parties analogiques (les plus sensibles aux imprécisions), puis les deux contrôleurs numériques (FPGA et PIC) sont assemblés pour la phase finale de test. Le FPGA est programmé et préalablement testé sur un circuit témoin avant d'être soudé, alors que le PIC est directement programmé sur le circuit au moyen de trois contacts prévus à cet effet.

Une fois les premiers tests fonctionnels passés, chaque implant doit subir un test d'endurance d'une semaine sur un banc d'essai qui simule une utilisation intensive du microstimulateur dans tous ses modes de fonctionnement. Ce banc d'essai, utilisant le CTF en mode automatique, reste toutefois trop simple et nécessite encore une inspection visuelle des signaux de stimulation à l'écran d'un oscilloscope.

Le milieu biologique dans lequel doit être implanté ce microstimulateur étant très humide et corrosif il est important d'isoler électriquement les circuits par une triple couche de vernis au polyuréthane adapté pour la protection des circuits électroniques. On remarquera que ce vernis n'est absolument pas adapté pour des applications médicales et qu'il devra être parfaitement sec et exempt de tout solvant avant d'être recouvert d'un matériau bio-compatible. Après 3 jours de séchage en étuve, l'implant ainsi protégé retourne pour une semaine sur le banc de test, mais cette fois-ci dans un bain de solution saline saturée à +40°C. Cette étape permet de s'assurer de l'intégrité de la pellicule protectrice.

Le dernière étape dans la préparation d'un implant consiste à l'enrober dans une silicone biomédicale, afin de protéger les tissus biologiques avec lesquels il sera mis en contact. En effet, sans ce matériau dit bio-compatible ou inerte, l'organisme rejette ce corps étranger en le bombardant d'anticorps. Ceci se matérialise généralement par de fortes infections autour de l'implant et des irritations cutanées sur le site de l'implantation. Pour assurer un enrobage uniforme, le microstimulateur est placé dans un moule où sera injecté la silicone liquide avant d'être cuit à 65°C pendant 6 heures. Sur la photographie

de la Figure 4.6, on peut voir un exemplaire du NSP enrobé de silicone transparent avec ses deux connecteurs plaqués or où viendront se brancher les électrodes de stimulation.

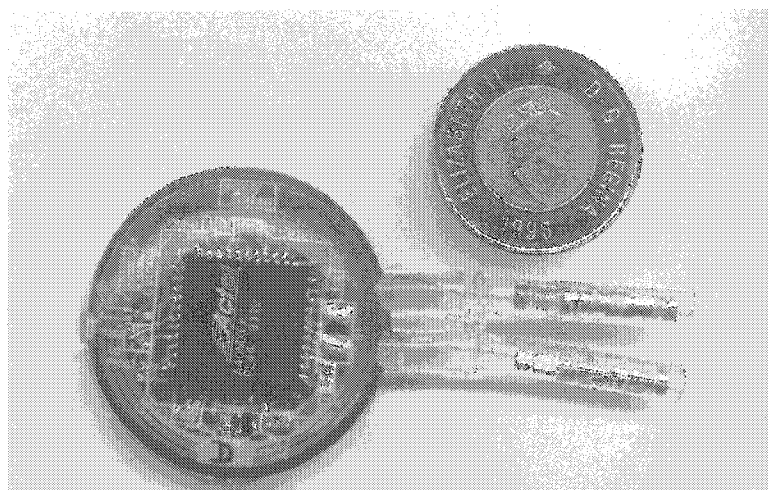


Figure 4.6 : Photographie du NSP prêt à être implanté

4.4.2 Les électrodes de stimulation

Nous avons jusqu'à présent principalement traité le cas des microstimulateurs et des contrôleurs externes, sans toutefois aborder le cas de la troisième partie essentielle d'un système de stimulation neurale : les électrodes de stimulation. La gaine d'une électrode se présente sous la forme de 2 anneaux de platine fixés à l'intérieur d'un cylindre de silicone. Ainsi, cette gaine s'enroule autour du nerf et les anneaux de platine assurent le contact électrique entre le nerf et le stimulateur. Le lien entre les 2 électrodes de la gaine et le microstimulateur est assuré par deux fils multibrins en acier inoxydable isolés par une couche de téflon. Pour maintenir l'électrode en place, deux techniques sont possibles : la première utilise des fils de suture qui encerclent la gaine de silicone (Figure 4.7.a) alors que la seconde se sert des avantages des alliages à mémoire de forme qui sont intégrés dans la structure de la gaine de silicone et remplacent les fils de suture (Figure

4.7.b). Cette dernière technique, mise au point par notre équipe de recherche [15], a l'avantage de simplifier le travail du chirurgien, en lui évitant de faire de points de suture dans une cavité peu accessible lors de l'opération. En plus, cela ne risquera pas de comprimer le nerf à cause d'un point de suture trop serré. La Figure 4.7.b présente une photographie d'une gaine à mémoire de forme où on peut reconnaître les 2 contacts de platine et les 3 anneaux d'alliage super-élastique (en noir).

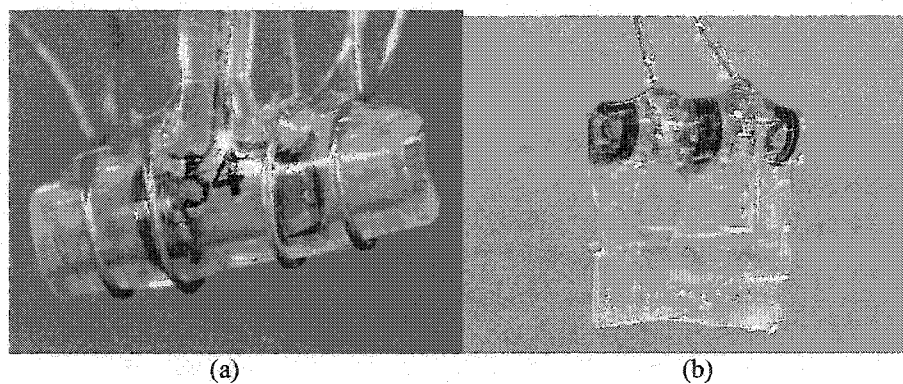


Figure 4.7 : Photographie d'une électrode à sutures (a) et d'une électrode à alliage super-élastique (b)

Finalement pour permettre la connexion directe des électrodes avec le microstimulateur, des connecteurs plaqués or et enrobés de silicone sont fixés aux extrémités des fils en aciers inoxydables provenant des contacts de platine. Ainsi, seul la silicone ou le téflon seront en contact direct avec les tissus vivants.

Une fois testés et protégés par un matériau bio-compatible, l'implant et ses électrodes sont prêts à être stérilisés au gaz avant d'être implantés chez un animal.

4.5 Protocole expérimental des tests en phase chronique

Notre collaboration avec le département d'urologie et le centre des ressources animales de l'Université McGill a donné lieu à l'étude en phase chronique de 6 chiens

paraplégiques implantés avec le nouveau microstimulateur (NSP). Cette étude a pour objectif de confirmer l'efficacité de stimulation sélective par blocage haute fréquence et d'observer les éventuels effets curatifs d'une stimulation permanente sur les problèmes d'hyper-réflexie de la vessie qui apparaissent à la suite d'une lésion de la moelle épinière. De plus, cette étude est dédiée à évaluer la fiabilité et la durabilité du nouveau microstimulateur dans un contexte réel d'utilisation.

4.5.1 Protocole chirurgical et implantation.

L'animal qui fera l'objet de l'étude sera préalablement choisi et examiné afin de s'assurer que son système urinaire soit intact et ne présente pas de malformations. Après une semaine d'observation, la chirurgie peut avoir lieu. En milieu stérile, l'animal est anesthésié et maintenu sous respirateur alors que ses fonctions vitales sont continuellement surveillées par de l'appareillage spécialisé. De plus, les pressions vésicale et urétrale sont enregistrées par un analyseur urodynamique connecté à un ordinateur de type PC. La chirurgie dédiée à l'implantation d'un stimulateur commence par une première incision au niveau de la vertèbre T10, afin de sectionner directement la moelle épinière, provoquant ainsi la paraplégie. Ensuite une seconde incision est faite au bas de la colonne vertébrale au niveau de la racine sacrée. Les nerfs sacrés S2 (gauche et droite) sont alors identifiés puis stimulés à l'aide d'une sonde à 2 pointes de platine et d'un stimulateur externe intégrant le même étage de stimulation que le microstimulateur, mais avec un contrôle directe des paramètres (Figure 4.8). Une électrode sera alors enroulée autour du nerf sacré S2 qui provoque la plus grande contraction de la vessie. Le microstimulateur est ensuite connecté à l'électrode, puis il est glissé sous la peau entre

l'hypoderme et les muscles. Le système de stimulation est testé une dernière fois avant de suturer l'incision. L'animal est placé dans une sorte de hamac pendant 3 ou 4 jours le temps qu'il s'habitue à sa nouvelle condition.

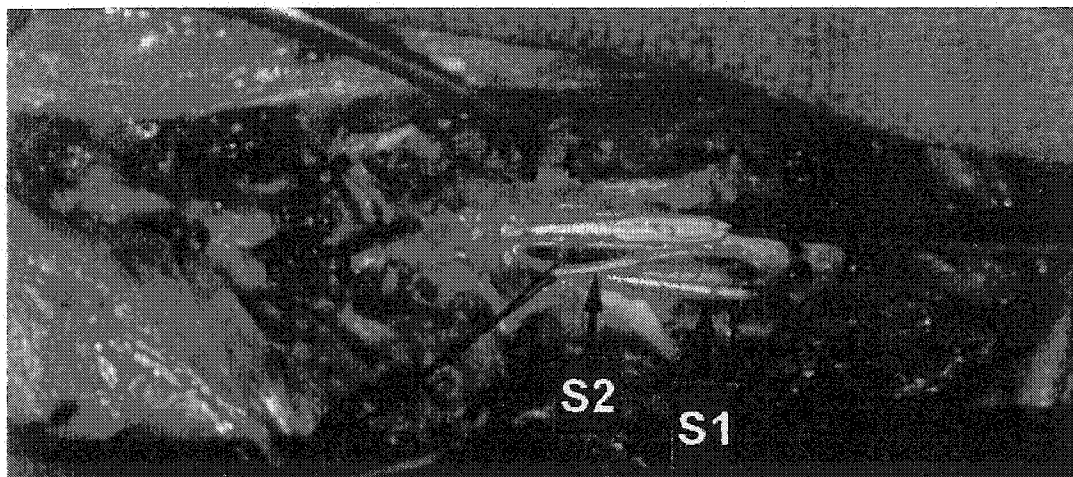


Figure 4.8 : Identification des nerfs de la racine sacrée

4.5.2 Suivi médical en phase chronique

L'étude en phase chronique s'étale sur une période de 8 mois. Tout au long de cette période, la santé des animaux est continuellement surveillée et la moindre infection ou complication vésicale, si elle y a lieu, est rapidement décelée et traitée. Immédiatement après la chirurgie (donc suite à la lésion) et pendant une période variable, pouvant atteindre 2 mois, la vessie entre dans une phase d'aréflexie durant laquelle aucune stimulation électrique ne sera appliquée. La vessie est simplement vidée de façon conventionnelle au moyen d'un cathéter placé dans l'urètre. Lorsque débute la phase d'hyperréflexie, la stimulation permanente est activée et la vessie de l'animal est vidée deux fois par jour au moyen de la stimulation sélective. Lors de la première stimulation,

un ensemble de paramètres de stimulation est choisi pour chaque animal afin de maximiser le volume évacué.

Les techniciens animaliers s'assurent donc quotidiennement du fonctionnement de la stimulation sélective et en évaluent l'efficacité. Si le volume d'urine résiduel après une stimulation est supérieur à 50% du volume total d'urine contenu dans la vessie, mais que des contractions sont toujours perceptibles, les paramètres de stimulation devront être changés par le chirurgien. Toutefois, si aucune contraction musculaire n'est ressentie, une intervention chirurgicale sera généralement requise, car cela signifie une défaillance du système de stimulation implanté.

Pour détecter le début et la fin des phases d'aréflexie et d'hyperréflexie, des cystomyogrammes (CMG) assistés par ordinateur sont effectués toutes les 2 semaines et permettent d'observer les pressions dans la vessie et dans l'urètre ainsi que l'activité électrique musculaire du detrusor. Une faible activité électrique musculaire permanente et une variation de la pression vésicale périodique sont caractéristiques de l'hyperréflexie. Ces CMG permettent aussi de contrôler plus précisément l'efficacité des paramètres de stimulation sélective et sont le seul moyen de vérifier de l'extérieur le fonctionnement de la stimulation permanente.

Les exemples de CMG présentés aux Figure 4.9 et Figure 4.10 illustrent les variations de pression et l'activité électrique musculaire de la vessie respectivement enregistrées lors d'une stimulation sélective et lors d'une stimulation permanente.

Une stimulation sélective se caractérise par une très forte activité électrique musculaire (EMG) du detrusor, une augmentation simultanée de la pression vésicale

(Pves) et urétrale (Pura), suivie d'une diminution de la pression urétrale liée au relâchement du sphincter qui provoque finalement la miction.

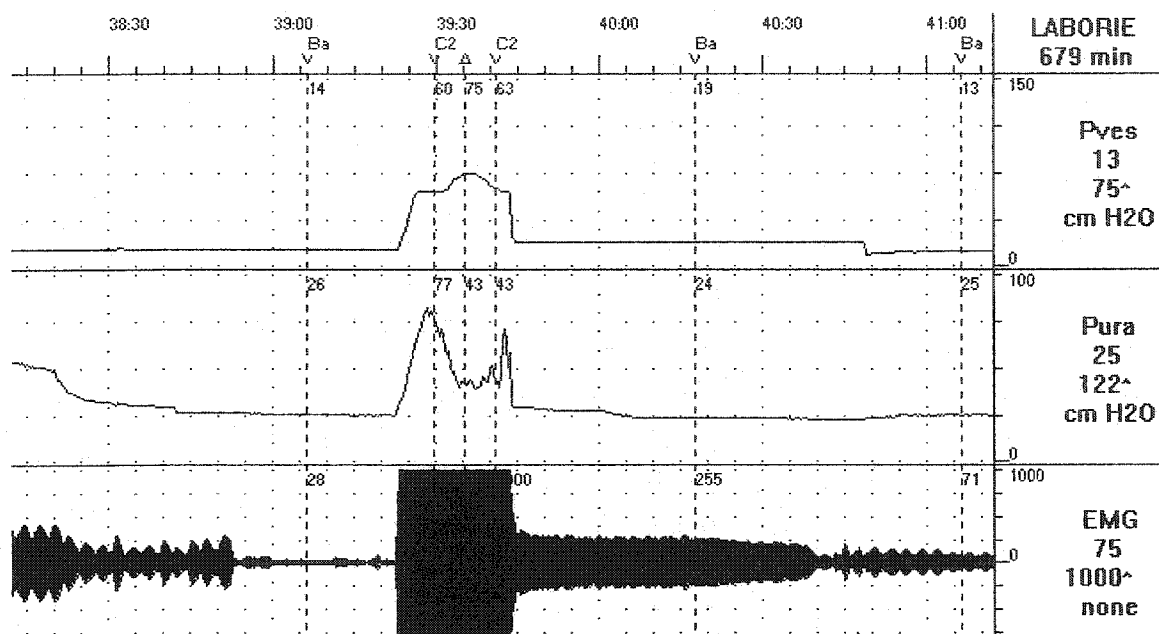


Figure 4.9 : Pression dans la vessie (Pves) et dans l'urètre (Pura) lors d'une stimulation sélective avec le NSP

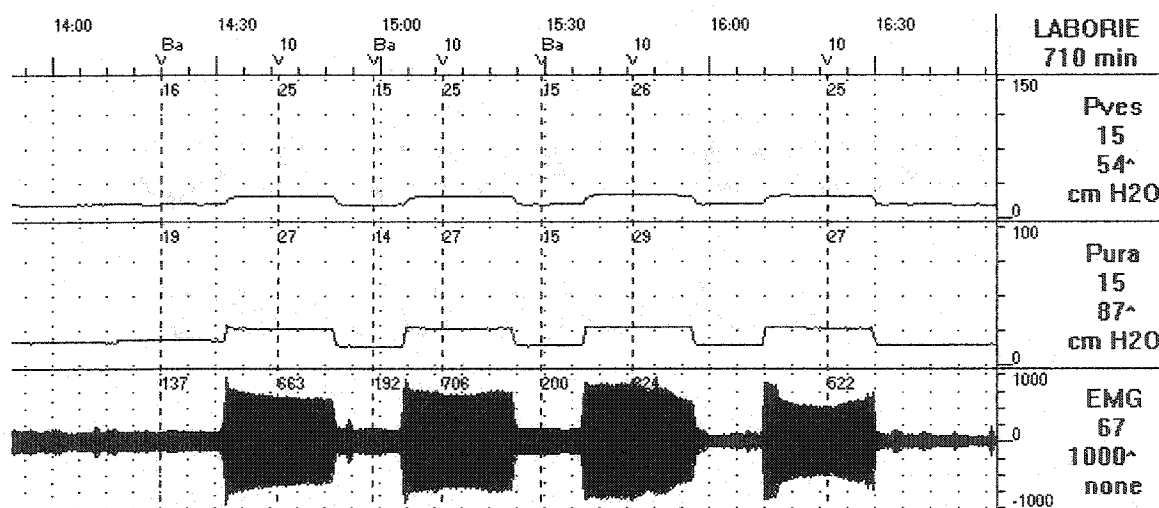


Figure 4.10 : Pression dans la vessie (Pves) et dans l'urètre (Pura) lors d'une stimulation permanente par le NSP

Lors d'une stimulation permanente, on constate une activité électrique beaucoup plus faible (puisque les courants de stimulation sont réduits) et de légères contractions du detrusor et du sphincter suffisamment faibles pour ne pas causer de miction.

4.6 Résultats expérimentaux et analyse de la fiabilité

Les CMG présentés à la Figure 4.9 et à la Figure 4.10 sont des courbes caractéristiques d'un implant considéré comme entièrement fonctionnel, puisqu'il est en mesure de provoquer une miction volontaire chez l'animal et de stimuler de façon autonome le nerf sacré. Tous les implants utilisés pour cette étude ont été entièrement fonctionnels durant le premier mois de leur implantation. Puis, leurs performances ont commencé à se dégrader, le contrôle externe devenait plus incertain et finalement chacun des microstimulateurs implantés a cessé de fonctionner et a dû être remplacé. Ainsi, nous présentons à la Figure 4.11 la chronologie des 8 exemplaires du NSP qui ont été utilisés pour l'étude en phase chronique chez l'animal. On constate en premier lieu que le système reste entièrement fonctionnel tant et aussi longtemps qu'il n'est pas implanté. Puis quelques mois après son implantation dans le corps de l'animal, le microstimulateur cesse complètement de fonctionner. Bien que cela ne soit pas précisé sur le diagramme, cet arrêt du fonctionnement est progressif et se manifeste principalement par 3 signes extérieurs : dans un premier temps, la stimulation permanente ne semble plus avoir aucun effet puis, quelques semaines après, la stimulation sélective commence à ne plus fonctionner que par intermittence et enfin, après quelques jours, le microstimulateur au complet devient inutilisable. Dans une telle situation, une chirurgie exploratoire est alors prévue pour inspecter l'implant et les électrodes. En effet, compte tenu des contraintes

mécaniques qui existent dans le corps de l'animal, la panne peut tout aussi bien venir du microstimulateur que d'un bris au niveau des électrodes.

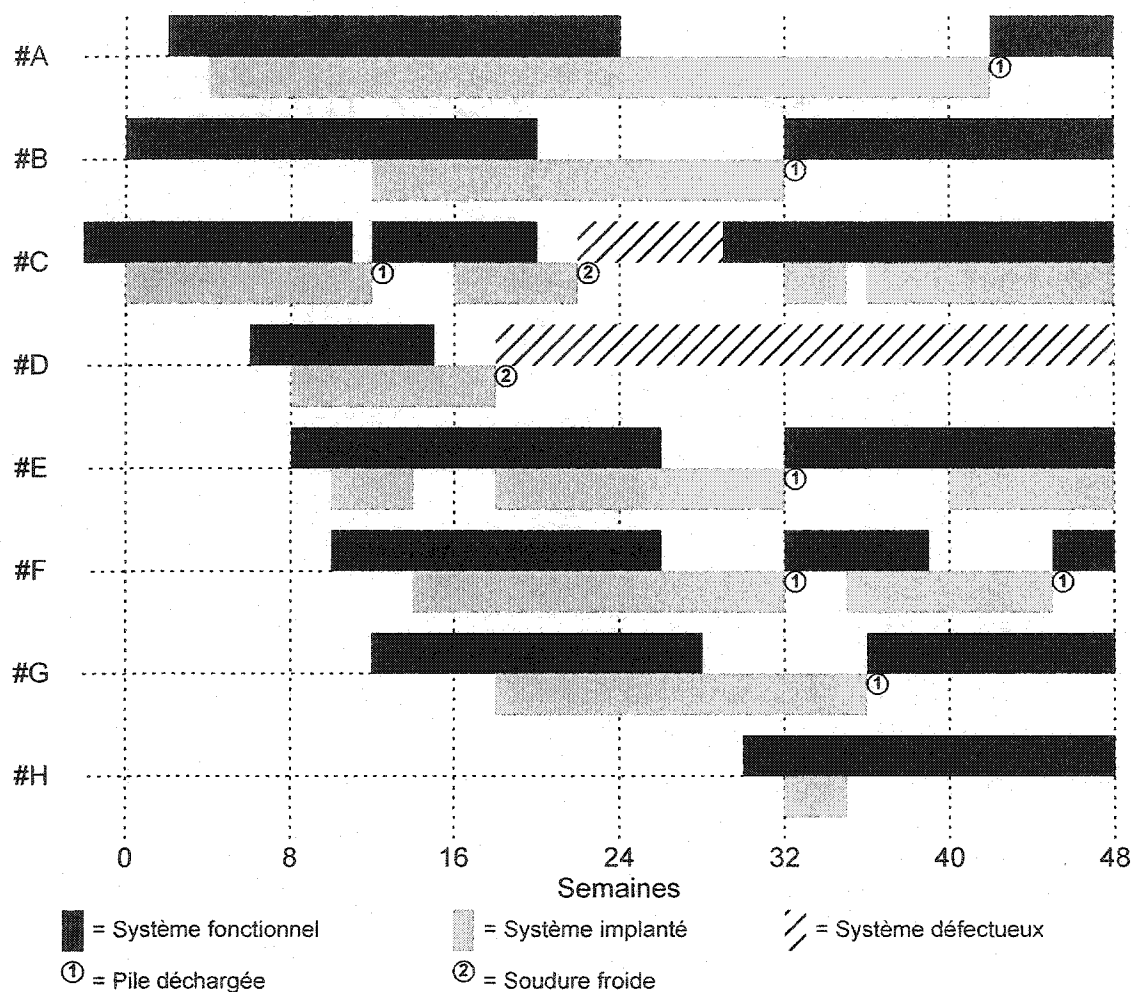


Figure 4.11 : Bilan de fiabilité des NSP pendant l'année d'évaluation

Dans 64% des cas, l'arrêt du fonctionnement été dû à une défaillance de la pile causée par une isolation électrique insuffisante, ou inadaptée compte tenu du milieu dans lequel elle se trouve implantée. Des traces de rouille ont été observées sous le vernis de polyuréthane à la frontière entre les deux pôles de la pile, cette dernière étant complètement déchargée. Immédiatement après le remplacement de la pile, les

microstimulateurs redevenaient entièrement opérationnels. Parallèlement, un implant de référence (#R) conservé au laboratoire et soumis à une utilisation normale, tout au long de l'étude, a permis de vérifier que le système reste fonctionnel pendant au moins 1 ans lorsqu'il n'est pas implanté. Dans deux cas particuliers, le dysfonctionnement venait d'une mauvaise soudure, appelée soudure froide, qui s'est manifestée avec le temps à la suite des différentes vibrations et variations de températures auxquelles ont été soumis ces microstimulateurs. Un de ces implants a pu être réparé, tandis que l'autre a subi des dommages irréversibles lors des manipulations de démoulage.

Enfin, dans 36% des cas, les électrodes ou le nerf de l'animal étaient aussi responsables du mauvais fonctionnement du système de stimulation. Il arrivait quelques fois que la soudure entre le contact de platine et les fils en métal inoxydable se brise ou encore que l'électrode se déplace et endommage le nerf.

D'autres informations importantes sur le bilan de fiabilité des NSP sont présentées au Tableau 4.7 et donnent un aperçu du taux d'utilisation de chaque microstimulateur.

Tableau 4.7 : Autres informations importantes sur la fiabilité des NSP

Identification de l'implant	Semaines fonctionnelles	Changement de pile	Implantations	Remarques sur les défaillances
#A	28 sur 44	1	1	Pile déchargée
#B	36 sur 48	1	1	Pile déchargée
#C	39 sur 50	2	3	Pile déchargée et soudure froide
#D	8 sur 38	1	1	Erreur de manipulation
#E	34 sur 40	1	3	Pile déchargée
#F	26 sur 38	2	2	Pile déchargée
#G	28 sur 36	1	1	Pile déchargée
#H	18 sur 18	0	1	Aucune
#R ¹	48 sur 48	0	0	Aucune
1 Implant de référence, conservé au laboratoire sur un banc de test pendant l'année d'étude				

Malgré les défaillances des microstimulateurs et le nombre d'interventions exploratoires, certains résultats médicaux ont pu être rassemblés pour valider l'efficacité de la stimulation sélective et vérifier l'influence de la stimulation permanente sur l'hyperréflexie. Le chronogramme de la Figure 4.12 résume les étapes de notre étude d'évaluation chez l'animal et fait ressortir les points saillants d'événements vécus des 6 sujets. En comparant ce chronogramme à celui de la Figure 4.11 on constate que la durée moyenne de la phase d'aréflexie est comparable à la durée de vie moyenne d'une pile implantée. Ainsi, lorsque les premiers hyperréflexes de la vessie apparaissent, les dernières ressources de la pile se trouvent épuisées et ne permettent plus qu'une utilisation de durée très limitée. Généralement, le remplacement de l'implant permet de prolonger l'étude, sans toutes fois pouvoir couvrir les huit mois avec un seul implant.

Les zones identifiées par "stimulation électrique" sur le chronogramme représentent les périodes pendant lesquels le microstimulateur implanté était entièrement fonctionnel, c'est-à-dire que plus de 75% du volume d'urine était évacué lors d'une stimulation sélective et que l'activité électrique créée par la stimulation permanente était identifiable lors d'une CMG. En dehors de ces périodes, soit une des fonctionnalités de l'implant n'était plus respectée, soit les techniciens animaliers nous rapportaient des problèmes d'utilisation du contrôleur externe ou un fonctionnement intermittent du stimulateur et dans certains cas, il n'y avait plus aucun signe de fonctionnement du microstimulateur, quelque soit la commande et la quantité d'énergie transmise.

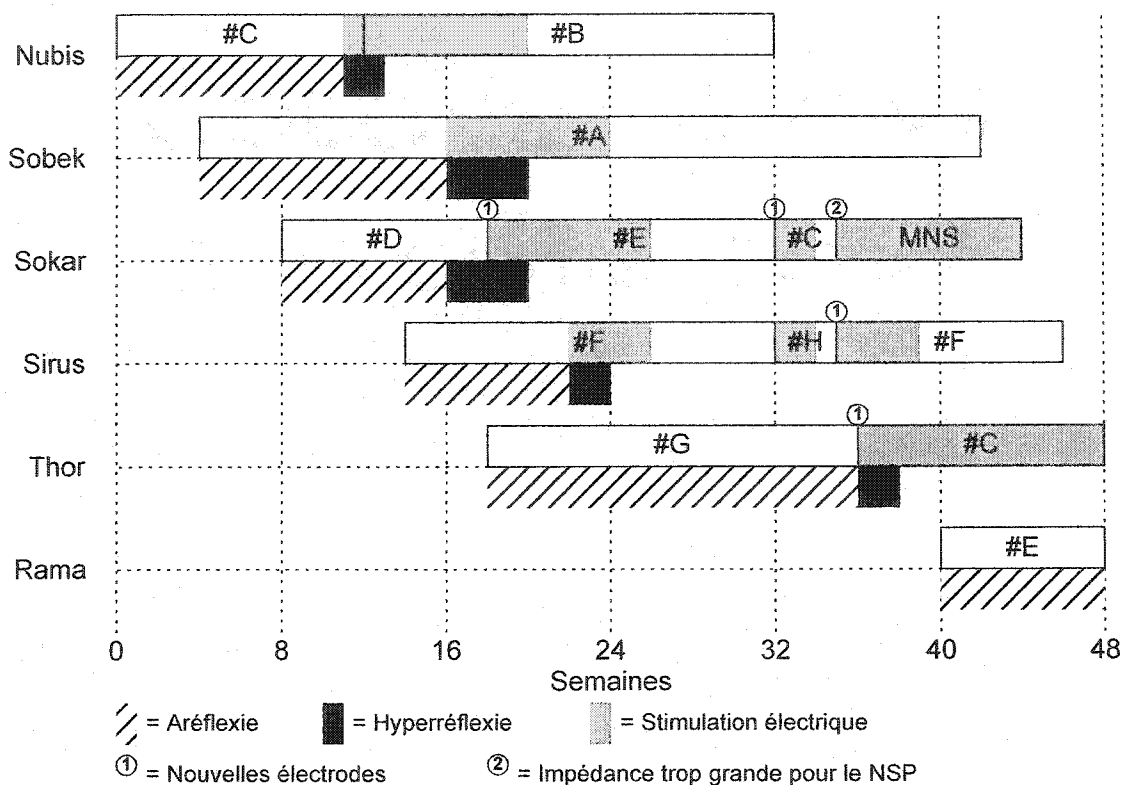


Figure 4.12 : Cheminement des travaux expérimentaux et principaux événements survenus

Toutefois, à un certain moment de l'étude, nous avons rencontré des problèmes avec l'amplificateur de puissance du contrôleur externe qui se déréglaient avec le temps et dont les transistors de sortie chauffaient anormalement. Ces problèmes ont perturbé l'étude, mais ils ont surtout permis de faire ressortir la fragilité du contrôleur externe et nous ont motivé à procéder au développement du module de stabilisation de la porteuse présenté au chapitre précédent. Ces travaux de recherche n'ont cependant pas abouti à une solution viable et réalisable à court terme, mais ils ont donné lieu à la mise au point de premières composantes d'une prochaine génération de contrôleurs externes. Devant le besoin urgent d'un contrôleur externe fonctionnel (5 sujets implantés), l'amplificateur existant a simplement subi de légères modifications pour stabiliser son fonctionnement et

accroître sa robustesse : les traces des signaux ont été élargies, les composants rapprochés l'un par rapport à l'autre et des dissipateurs de chaleur ont été placés sur les transistors.

En plus des problèmes du microstimulateur, du contrôleur externe et des électrodes, viennent se rajouter les traumatismes du nerf qui à force d'interventions peut perdre de sa sensibilité ou encore peut changer de propriétés électriques. Dans un des cas (le sujet Sokar), il a même été nécessaire de remplacer un NSP par un MNS classique pour pouvoir terminer l'étude, car l'impédance du nerf était devenue trop élevée pour la source de courant du NSP qui saturait en tension bien avant d'atteindre le courant de stimulation désiré. En effet, en voulant minimiser la consommation de puissance pour le NSP, nous avons abaissé la tension d'alimentation de 5 V à 3.3V, ce qui a réduit la plage de tension de sortie de la source de courant du microstimulateur.

En dépit de tous ces ajustements, les résultats médicaux de cette première étude ont démontré une très nette réduction de la phase d'hyperréflexie sur tous les sujets traités par une stimulation permanente. En effet, sur le chronogramme de la Figure 4.12, on note une disparition complète des hyperréflexes de la vessie en moyenne 2 semaines après le début du traitement par la stimulation permanente. Dans l'étude précédemment faite par notre équipe de recherche sur l'efficacité de la stimulation sélective avec le MNS sur 10 autres sujet [8], la durée moyenne de la phase d'hyperréflexie atteignait 2 mois. Cette phase peut parfois atteindre 4 à 6 mois sur des sujets sans stimulation électrique (sélective ou permanente). En plus de disparaître rapidement, il s'est avéré que cette hyperréflexie ne soit jamais réapparu pour aucun des sujets traités, contrairement à ce qui se produit pour des sujets sans stimulation électrique permanente.

4.7 Conclusions

Les résultats présentés dans ce chapitre montrent que la complexité dans la conception d'un système médical dédié réside autant dans la réalisation d'un dispositif électronique fiable que dans l'adaptation de ce système au milieu dans lequel il sera utilisé et à l'utilisateur qui s'en servira. Ainsi, le NLB permet une mesure de l'impédance relativement précise mais nécessite encore beaucoup trop de manipulations de la part du patient. La fiabilité du NSP demeure encore très dépendante de la pile qui elle même demanderait une plus grande attention quand à son isolation contre les anticorps des milieux vivants. Le vernis choisi est suffisant pour protéger les circuits électroniques du microstimulateur, mais la pile au lithium faite de matériaux inoxydables représente une surface moins adhérente pour le vernis et les réactions chimiques entre le vernis et les fluides corporels de l'animal sont sûrement facilitées par la différence de potentiel permanente qui existe autour de la pile.

Les expérimentations en phase chronique nous ont permis de déceler les faiblesses du système de stimulation lors de son utilisation dans des conditions réelles. Ces problèmes ne sont pas seulement venus de microstimulateurs implantés, mais aussi de leur contrôleur externe et des électrodes. Nous avons dû en particulier refaire le contrôleur externe pour le rendre plus robuste et plus stable et nous avons dû optimiser certaines étapes dans la soudure à l'arc des fils inoxydables sur les contacts de platine des électrodes.

Finalement, les résultats médicaux obtenus démontrent l'utilité de la stimulation permanente pour le traitement de l'hyperréflexie de la vessie chez les sujets

paraplégiques. Les difficultés techniques rencontrées pendant l'étude n'ont pas permis de déterminer le lien exact qui existe entre les paramètres de la stimulation permanente (amplitude, fréquence, largeur d'impulsions, durée active et durée inactive) et la durée de la phase d'hyperréflexie. Lors de l'étude, seul l'amplitude était modifiée au cours du temps et en fonction de l'animal, les autres paramètres restant inchangés (Amp=50 μ A-200 μ A, Freq=20Hz, PW=200 μ s, T_{on}=18s et T_{off}=10s).

CONCLUSION

Les travaux présentés dans ce mémoire constituent une suite logique à l'effort de recherche apporté depuis ces 10 dernières années par l'équipe PolySTIM dans le domaine de la stimulation électrique fonctionnelle. Les progrès de la microélectronique permettent de nos jours de réaliser rapidement, à partir de circuits intégrés programmables disponibles commercialement, des dispositifs fonctionnels implantables sans avoir recours à des technologies plus coûteuses et délicates comme les prédiffusés. Ainsi nous avons pu concevoir et tester en peu de temps deux microstimulateurs sur des circuits imprimés dont les dimensions sont tout à fait comparables à celles des stimulateurs cardiaque (pacemaker) qui sont couramment utilisés chez l'Homme.

Malgré les nombreuses recherches dans le domaine de la réhabilitation des fonctions vésicales chez les paraplégiques, il n'existe actuellement qu'un seul système électronique implantable (*VOCARE® bladder system*) qui ait été homologué pour le marché nord américain. Ce dispositif comporte cependant de nombreux effets secondaires et utilise une technologie ancienne. Il a toutefois bénéficié de l'avantage du premier venu et a commencé les étapes d'homologation il y a plus de 10 ans. Mis à part cet implant commercial, plusieurs équipes font de la recherche dans le domaine, mais très peu de solutions effectivement réalisées sont comparables avec le dernier stimulateur que nous proposons.

La première génération de microstimulateurs sélectifs, le MNS, a été introduite en 1998 et a permis, sur une étude de 3 ans en phase chronique, de démontrer l'efficacité de

la stimulation sélective par blocage à haute fréquence pour la miction volontaire chez les paraplégiques.

Contributions du mémoire

L'introduction de la nouvelle génération du microstimulateur (NSP) avec sa fonction de stimulation permanente pour le traitement de l'hyperréflexie, prouve qu'il est possible de rétablir l'ensemble des fonctionnalités du système urinaire avec un seul et même implant. Enfin, le microstimulateur à lien inductif bidirectionnel (NLB) vient combler les derniers besoins du système de stimulation en assurant un suivi permanent sur l'état des tissus stimulés lors d'implantation à long terme.

Ces deux dernières générations de stimulateurs constituent donc un apport important dans le domaine de la réhabilitation vésicale, puisqu'ils permettent aussi bien une miction volontaire sans effets secondaires (au vu des résultats actuels), un traitement de l'hyperréflexie du muscle de la vessie et une source d'information nécessaire sur l'évolution de l'état du nerf stimulé lors d'une implantation de longue durée.

Les contributions de nos travaux de recherche ne se limitent pas à l'introduction de ces nouveaux microstimulateurs. En effet, les études expérimentales sur l'animal menées parallèlement au développement des implants ont permis d'acquérir une meilleure connaissance des contraintes électriques et mécaniques appliquées aux dispositifs électroniques implantés dans un corps vivant. L'identification de problèmes comme celui de l'isolation insuffisante de la pile ou encore celui de la fragilité des soudures des fils inoxydables sur les contacts de platine des électrodes, permettra de développer de nouveaux procédés qui contribueront à accroître la fiabilité et la robustesse

des systèmes électroniques que nous réalisons. De plus, une telle expérience renforce l'expertise de notre équipe de recherche multidisciplinaire et représente une source d'informations essentielle pour le développement des futurs implants.

Finalement, les efforts mis sur la conception de nouveaux contrôleurs externes conviviaux démontrent l'importance de concevoir des systèmes électroniques qui sont non seulement performants et stables, mais qui doivent aussi être adaptés à des manipulations quotidiennes par des usagers non initiés à l'électronique.

Recommandations

Les travaux expérimentaux effectués par le NSP ont permis d'identifier 3 problèmes sur ce nouveau stimulateur pour lesquels nous émettons les recommandations suivantes :

- l'isolation électrique insuffisante de la pile réduit la durée de fonctionnement du microstimulateur une fois implanté. Cette faiblesse pourrait être corrigée en analysant précisément les fluides corporels mis en contact avec le vernis protecteur qui se trouve endommagé. Une solution simple consiste à trouver un nouveau vernis plus résistant, et une meilleure solution serait de sceller l'implant dans un boîtier en titane étanche, mais en faisant attention à la disposition de l'antenne en se servant de vias étanches.
- la stimulation sélective s'arrête définitivement lorsque la pile est déchargée. Nous proposons de séparer autant que possible les circuits des deux stimulations sélective et permanente, chacun ayant leur propres générateurs de stimuli (CNA, source de courant et commutateur de courant).

- la tension d'alimentation de 3.3V, utilisée pour la stimulation sélective, réduit la plage dynamique de la tension de sortie du microstimulateur et provoque la saturation de l'amplificateur opérationnel utilisé pour la source de courant lorsque l'impédance du nerf devient trop élevée. Il serait nécessaire de modifier le circuit pour permettre d'alimenter l'implant à une tension de 5V pour la stimulation sélective et conserver une tension de 3V pour la stimulation permanente.

En plus des difficultés rencontrées avec le microstimulateur, certains autres problèmes provenaient du contrôleur externe qui se décalibre fréquemment et dont certains transistors surchauffent. Des solutions à ces problèmes ont été proposés au chapitre 3 avec le module de stabilisation de la porteuse par contrôle numérique. Toutefois, ces modifications n'ont pu être apportées sur le contrôleur externe actuel, car les composants disponibles ne permettaient pas d'atteindre des vitesses de commutation de 20MHz. De tels limitations pourraient être évitées à l'avenir si la fréquence de la porteuse était réduite autour de 10MHz. De plus, le remplacement de la modulation ASK, entrepris par notre équipe, devrait améliorer les performances énergétiques du lien inductif pour permettre de transférer plus d'énergie à l'implant.

La fonctionnalité des principes de stimulation sélective et permanente étant maintenant vérifiée, il resterait à corriger les problèmes dont nous avons discuté dans cette conclusion. Ensuite, il faudrait envisager la réalisation d'un circuit intégré dédié de taille réduite en comparaison des implants actuels. Toutefois, les dimensions globales de l'implant resteraient dictées par la surface de l'antenne inductive qui ne peut pas être trop petite.

BIBLIOGRAPHIE

- [1] AHLUWALIA, A., MAGGI, C.A., SANTICIOLI, P., LECCI A. et GIULIANI, S. (1994). Characterization of the capsaicin sensitive component of the cyclophosphamide-induced inflammation in the rat urinary bladder. Br. J. Pharm., 111, 1017.
- [2] ARABI, K. (1994). Conception d'un microstimulateur neuromusculaire destiné à la récupération des dysfonctions urinaires. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.
- [3] ARABI, K. et KAMINSKA, B. (1996). A new technique to monitor the electrode and lead failures in implantable microstimulators and sensors. EMBS Conference, article 786.
- [4] ARABI, K. et SAWAN, M. (1996). Implantable multiprogrammable microstimulator dedicated to bladder control. Medical and Biological Engineering and computing, 48-51.
- [5] BARADA, H., SAWAN, M., HASSOUNA, M., TU, L.M. et ELHILALI, M.M. (1995). Sphincter fatigue strategy for bladder control : preliminary chronic results in an animal model. Conf. EMBS, article 437.
- [6] BARATA, R., ICHIE, M., HWANG, S. et SOLOMONOW, M. (1989). Orderly stimulation of skeletal muscles motor units with tripolar nerve cuff electrode. IEEE Transaction on Biomedical Engineering, 36, 836.
- [7] BLAIVAIS, J.G. (1982). The neurophysiology of micturition : a clinical study of 550 patients. J. Urol., 958-962.
- [8] BOYER, S., SAWAN, M., ABDEL-GAWAD, M., ROBIN, S. et ELHILALI, M. (2000). Implantable selective stimulator improve bladder voiding : design and chronic experiments in dogs. IEEE Transactions on Rehabilitation Engineering, 8, 764-470.

- [9] BRADLEY, W.E., ROCKSWOLD, G.L., TIMM, G.W. et SCOTT, F.B. (1976). Neurology of micturition. J. Urol., 115, 481.
- [10] BRINDLEY, G.S. (1988). The action of parasympathetic and sympathetic nerve in human micturition, erection, and seminal emission and their restoration in paraplegic patients by implanted electrical stimulators. Proc. R. Soc. London, 235, 111-120.
- [11] BRINDLEY, G.S. (1994). The first 500 patients with sacral anterior root stimulation implants. Paraplegia, 32, 795-805.
- [12] BRINDLEY, G.S. et CRAGGS, M.D. (1980). A technique for anodally blocking large nerve fibers through chronically implanted electrodes. J. Neurol. Neurosurg. And Psych., 43: 1083-1090.
- [13] BRINDLEY, G.S., POLKEY, C.E. et RUSHTON, D.N. (1982). Sacral anterior root stimulation for bladder control in paraplegia. Paraplegia, 20, 365-381.
- [14] CRAMPON, M.A. (1999). Conception et réalisation d'électrodes neuronales dédiées à des stimulateurs électroniques implantables. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.
- [15] CRAMPON, M.A., SAWAN, M., BRAILOVSKI, V. et TROCHU, F. (1998). New nerve cuff electrode based on a shape memory alloy armature. Proceeding of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 20, 2556-9.
- [16] CREASY, G.H.(1993). Electrical stimulation of sacral roots for micturition after spinal cord injury. Urol. Clin. North Am., 20. 505.
- [17] DOBELLE, W.H. et MLADEJOVSKY, M.G. (1974). Phosphene produced by electrical stimulation of human occipital cortex and their application to the development of a prosthesis for the blind. J. Physiol., 243, 553-576.
- [18] DONALDSON, N. et PERKINS, T.A. (1983) Analysis of resonant coupled coils in design of radio frequency transcutaneous links. Med. Biol. Eng. Comput., 21, 612-627.

- [19] DONFACK, C. (2000). Caractérisation de contacts électrodes-tissus pour les stimulateurs neuro-musculaires implantables. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.
- [20] DONFACK, C., SAWAN, M. et SAVARIA, Y. (2000). Implantable measurement technique dedicated to the monitoring of electrode-nerve contact in bladder stimulators. Medical and Biological Engineering and Computing, **38**, 465-8.
- [21] FAM, B.A., ROSSIER, A.B. et BLUNT, K. (1978). Experience in the urologic management of 120 early spinal cord injury patients. J. Urol., **119**, 485.
- [22] GEIRSSON, G., FALL, M. et SULLIVAN, L. (1995). Clinical and urodynamic effects of intravesical capsaicin treatment in patients with chronic traumatic spinal detrusor hyperreflexia. J. Urol., **154**, 1825.
- [23] GREATBATCH, W. et HOLMES, C. (1991). History of implantable devices. IEEE Eng. Med. Bio., 38-41.
- [24] GROAT, W.C. (1995). Mechanisms underlying the recovery of lower urinary tract function following spinal cord injury. Paraplegia, **33**, 493.
- [25] GRUNEWALD, V., BHADRA, N., CREASEY, G.H. et MORTIMER, J.T. (1998). Functional conditions of micturition induced by selective sacral anterior root stimulation: Experimental results in a canine animal model. World. J. Urol., **16**, 329-336.
- [26] HABIB, H.N. (1967). Experience and recent contribution in sacral nerve stimulation for voiding in both human and animal. Brit. J. Urol., **39**, 73-83.
- [27] HABLER, H.J., JANIG, W. et KOLTZENBURG, M.(1990). Activation of unmyelinated afferent fibers by mechanical stimuli and inflammation of the urinary bladder in the cat. J. Physiol., **425**, 545.
- [28] HALD, T., AGRAWAL, G. et KANTREWITZ, A. (1966). Studies in stimulation of the bladder and its motor nerves. Surgery, **60**, 848.
- [29] HALVERSTED, D.B. (1971). Electrical stimulation of the human bladder, 3 years later. J. Urol., **106**, 673.

- [30] HAMBRECH, F.T., BAK, M., GIRVIN, J.P., KUFFA, C.V., LOEB, G.E. et SCHMIDT, E.M. (1990). Visual sensations produced by intracortical microstimulation of the human occipital cortex. Med. & Bio. Eng. & Comput, 28, 257-259.
- [31] HASSOUNA, M., LI, J.S. et ELHILALI, M. (1994). Dog as an animal Model for Neurostimulation. Neurourologie and Urodynamics, 13, 159-167.
- [32] HAUGLAND, M. (1996). A flexible method for fabrication of nerve cuff electrodes. IEEE/EMBS Proceedings.
- [33] HOLMQUIST, B. (1968). Electromicturition by pelvic nerve stimulation in dogs. J. Urol. Nephrol., 2, 1.
- [34] HOLMQUIST, B. et OLIN, T. (1968). Electrostimulation in male dogs at pelvic nerve stimulation: a urethrocytographic study. J. Urol. Nephrol., 2, 155.
- [35] JONES, U. et TANAGHO, E.A. (1975). Spinal cord stimulation versus detrusor stimulation. A comparative study in dogs. Invest. Urol., 13, 171.
- [36] JONES, U. et TANAGHO, E.A. (1976). Studies on vesicourethral reflexes. II Urethral sphincter responses to spinal cord stimulation. Invest. Urol., 13, 278.
- [37] KAPLAN, A.C., CHANCELLOR, M.B. et BLAIVAIS, J.G. (1991). Bladder and sphincter behaviour in patients with spinal cord lesions. J. Urol., 146, 113.
- [38] KHALAF, I.M., TOPPERCER, A. et ELHILALI, M.M. (1979). Urethral pressure responses to detrusor stretch. Invest. Urol., 17, 141.
- [39] KO, W.H., LIANG, S.P. et FUNG, C.D.F. (1977). Design of radio-frequency powered coils for implant instruments. Med. Biol. Eng. Comput., 15, 634-640.
- [40] LI, J.P., HASSOUNA, M., SAWAN, M., DUVAL, F. et ELHILALI, M.M. (1992). Electrical stimulation induced sphincter fatigue during voiding. J. Urol., 148, 949-951.
- [41] LI, J.P., HASSOUNA, M., SAWAN, M., DUVAL, F. et ELHILALI, M.M. (1995). Long-term effect of sphincteric fatigue during bladder neurostimulation. J. Urol., 153, 242.

- [42] LINDSTORM, S., MAZIERES, L.A. et JIANG, C. (1990). The bladder C-fiber reflex : an experimental study in the cat. Neurol. Urodyn., 9, 365.
- [43] MADERBACHER, H.G., KOFLER, A., FISCHER, J., STRASSER, F., HARING, B. et SPANUDAKIS (1999). Early temporary and late permanent loss of electromicturition through an anterior sacral root stimulator (Brindley)- A 12 years experience. European Urology, XIV Congress of the European Association of urology, Abstract 60.
- [44] McDONAGH, R.P., FORSTER, D.M. et THOMAS, D.G. (1990). Urinary incontinence in spinal cord injury patients following complete sacral posterior rhizotomy. Br. J. Urol., 66, 618-622.
- [45] MEDTRONIC (1999). Interstim® therapy for urinary control. Medtronic Neurological Inc., Minneapolis.
- [46] MOUINE, J., WEI, M., FONTAINE, R. et DUVAL, F. (1995). A full custom sixteen-channel cochlear implant microstimulator. Conférence EMBS, article 126.
- [47] NASHOLD, B.S., FRIEDMAN, H., GLENN, J.F., GRIMES, J.H., BOVY, W.F. et AVERY, R. (1972). Electrostimulation in paraplegics. Arch. Surg., 104, 195.
- [48] NEUROCONTROL CORPORATION (1998). VOCARE® bladder system, implantable functional neuromuscular stimulator. Neurocontrol Corporation, Ohio.
- [49] RIJKHOFF, N. J., WIJKSTRA, H., VAN KERREBROCK, P.E.V. et DEBRUYNE, F.M.J. (1997). Selective detrusor activation by electrical sacral nerve root stimulation in spinal cord injury. J. Urol., 157, 1504-1508.
- [50] RIJKHOFF, N.J., KOLDEWIJN, P.E., VAN KERREBROCK, P.E.V., DEBRUYNE, F.M.J. et WIJKSTRA, H. (1994). Acute animal studies on the use of an anodal block to reduce urethral resistance in sacral root stimulation. IEEE Trans. Rehab. Eng., 2, 92-94.
- [51] RIVAS, D.A., et CHANCELLOR, M.B. (1995). Neurogenic vesical dysfunction. Urol. Clin., North Am., 22, 579.

- [52] ROBIN, S. (1998). Réalisation et test d'un système implantable dédié à la stimulation neurale sélective. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.
- [53] ROBIN, S., SAWAN, M., ABDEL-GAWAD, M., ABDEL-BAKY, T.M. et ELHILALI, M.M. (1998). Implantable stimulation system dedicated for neural selective stimulation. Med. and Bio. Eng. and Comput.
- [54] ROY, M. (1999). Conception et réalisation d'un prototype de la partie implantable d'un stimulateur visuel cortical. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.
- [55] SAUERWEIN, D.H., DOMURATH, B.K.H. et KUTZENBERGER, J.F. (1999). Experience with sacral deafferentation and implantation of an anterior root stimulator following in 294 spinal-cord injury patients. J. Urol., 161, 274.
- [56] SAWAN, M., DUVAL, F., HASSOUNA, M. et ELHILALI, M.M. (1993). A new bladder stimulator – Hand-held controller and miniaturized implant : preliminary results in dogs. Biomedical Instrumentation and Technology, 143-149.
- [57] SCHMIDT, R.A., BRUSCHINI, H. et TANAGHO, E.A. (1979). Urinary bladder and sphincter responses to stimulation of dorsal and ventral sacral roots. Invest. Urol., 16, 300.
- [58] SCHUMACHER, S., BROSS, S., SCHEEPE, J.R., SIEF, C., JUNEMANN, K.P. et ALKEN, P. (1999). Extradural cold block for selective stimulation of the bladder: Development of a new technique. J. Urol., 161, 950-954.
- [59] SCHURCH, B., HAURI, D., RODIC, B., CURT, A., MEYER, M. et ROSSIER, A.B. (1996). Botulinum-A Toxin as Treatment of Detrusor-Sphincter Dyssynergia : Prospective Study in 24 Spinal Cord Injury Patients. Journal of Urology, 155, 1023-1030.
- [60] SCHURCH, B., KNAPP, P.A., JEANMOND, D., RODIC, B. et ROSSIER, A.B. (1998). Does sacral posterior rhizotomy suppress autonomic hyper-reflexia in patients with spinal cord injury? Brit. J. Urol., 81, 73-82.

- [61] SCHURCH, B., RODIC, B. et JEANMOND, D. (1997). Posterior sacral rhizotomy and intradural anterior sacral root stimulation for treatment of the spastic bladder in spinal cord injury. J.Urol., 157, 610-614.
- [62] SCHURCH, B., STOHRER, M., KRAMER, G., SCHMID, D.M., GAUL, G. et HAURI, D. (2000). Botulinum-A Toxin for Treating Detrusor Hyperreflexia in Spinal Cord Injured Patients: New Alternative to Anticholinergic Drugs? Preliminary Results. Journal of Urology, 164, 692-698.
- [63] SCOTT, F.B., QUESADA, E.M., CARDUS, D. et LASKOURSKI, T. (1965). Electronic bladder stimulation, dog and human experiments. Invest. Urol., 3, 231.
- [64] SHAKER, H., TU, L.M., KALFOPOULOS, M., HASSOUNA, M., DION, S., et ELHILALI, M.M. (1998). Hyperreflexia of the urinary bladder : Possible role of the efferent function of the capsaicin sensitive primary afferents. J. Urol., 160, 2232.
- [65] SHAKER, H.S., TU, L.M., ROBIN, S., ARABI, K., HASSOUNA, M., SAWAN, M. et ELHILALI, M.M. (1998). Reduction of Bladder Outlet Resistance by Selective Sacral Root Stimulation Using High-Frequency Blockade in Dogs: Acute study. J. Urol., 16, 901.
- [66] SMITH, B., PECKHAM, P.H. et KEITH, M.W. (1987). An externally powered multichannel implantable stimulator for versatile control of paralysed muscle. IEEE Transactions on Bio-Medical Engineering, 34, 499-508.
- [67] SOLOMONOW, M. (1984). External control of neuromuscular system. IEEE transactions on Biomedical Engineering, 31, 752.
- [68] STOHRER, M: (1990). Alteration of the urinary tract after spinal cord injury: diagnosis, prevention, and therapy of the sequelae. World J. Urol., 7, 205.
- [69] TANAGHO, E.A. et SCHMIDT, R.A. (1991). Restoration of voiding in chronic urinary retention using neuroprosthesis. World J. Urol., 9, 142.
- [70] TANENBAUM, A.S. (1989). Computer networks. NJ, Prentice-Hall.
- [71] TERMAN, F.E. (1943). Radio engineering handbook. New York, McGraw-Hill.

- [72] THUROFF, J.W., SCHMIDT, R.A., BAZEED, M.A. et TANAGHO, E.A. (1983). Chronic stimulation of sacral roots in dogs. Eur. Urol., 9, 102-108.
- [73] TIM, G.W. et BRADLEY, W.E. (1969). Electrostimulation of the urinary detrusor to effect contraction and evacuation. Invest. Urol., 6, 562.
- [74] WELD, K.J., GRANEY, M.J. et DMOCHOWSKI, R.R. (2000). Clinical significance of detrusor sphincter dyssynergia type in patients with post-traumatic spinal cord injury. Urology, 56, 565-568.
- [75] WOODFORD, B.J., CARTER, R.R., McCREERY, D.B., BULLARA, L.A. et AGNOW, W.F. (1996). Histopathological and physiologic effects of chronic implantation of microelectrodes in sacral spinal cord of the cat. J. Neuropathol. Exp. Neurol., 55, 982.
- [76] YALLA, S.V., ROSSIER, A.B., et FAM, B.A. (1976). Dyssynergic vesicourethral responses during bladder rehabilitation. J. Urol., 115, 575.

ANNEXE A

SCHÉMAS ET CODES DU NLB

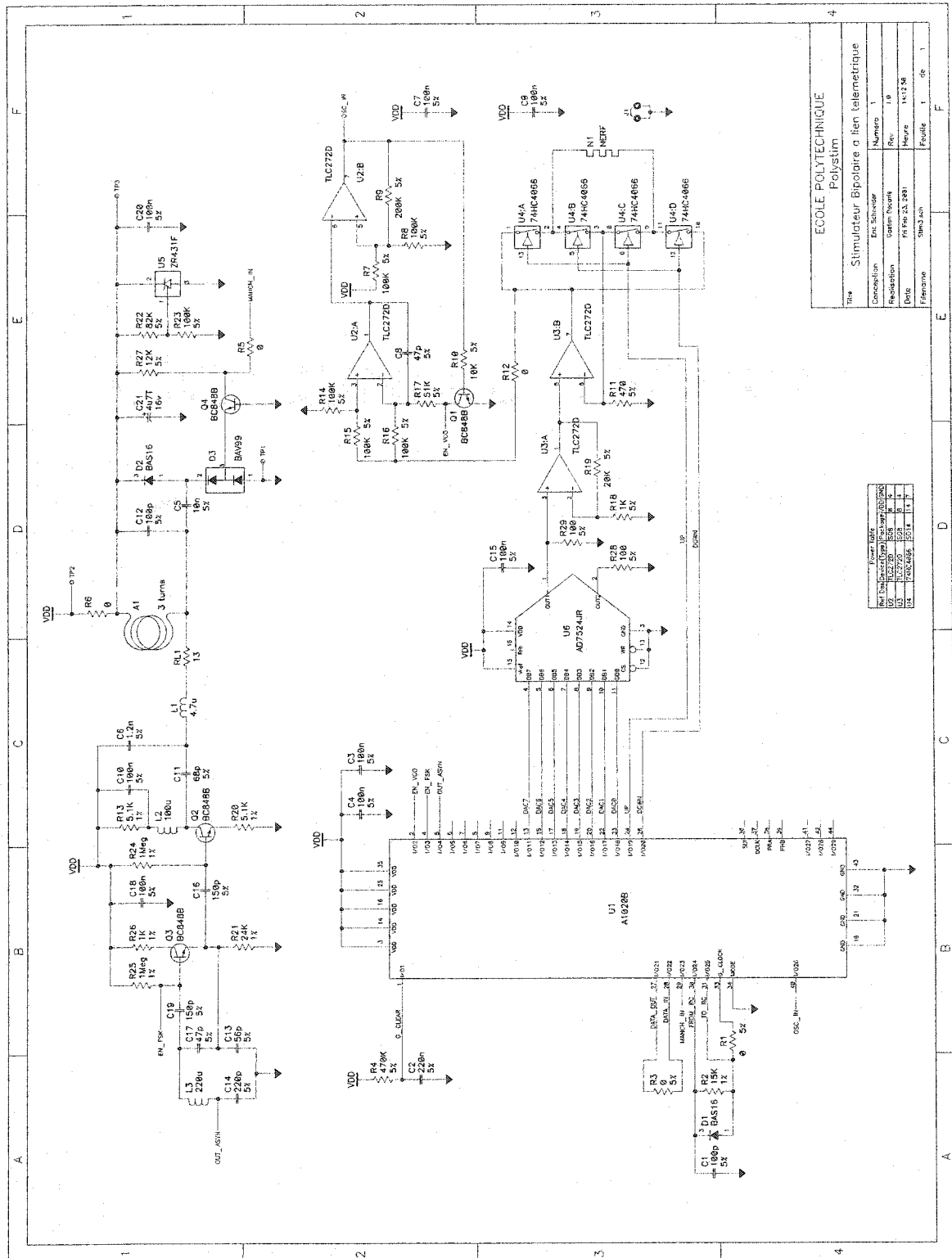


Figure A.1 : Schéma électrique du NLB

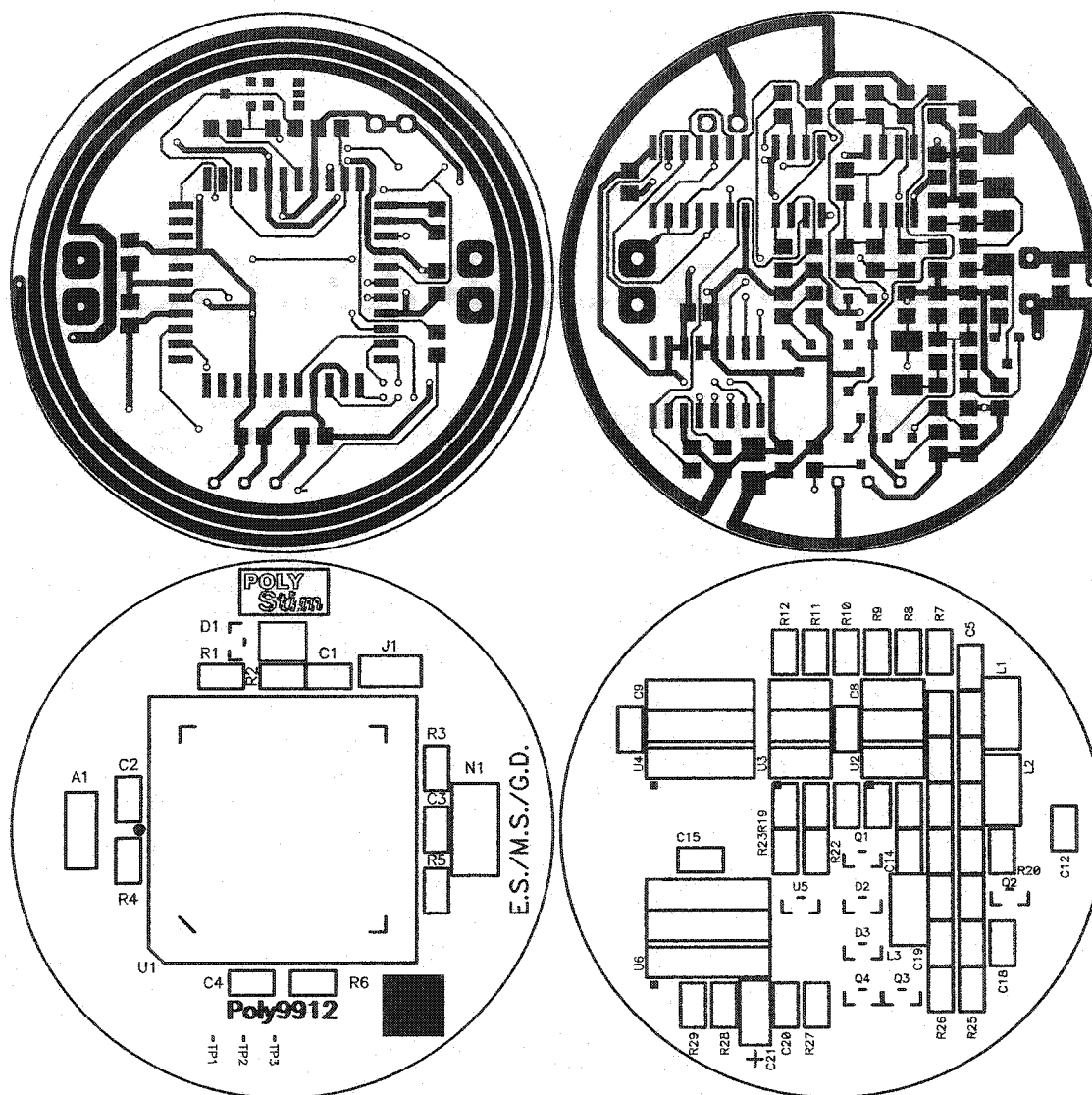


Figure A.2 : Dessin des masques à l'échelle 15/10 du NLB

Numéro de référence	Boîtier	Identificateur	Description	Fabricant
A40MX04	PLCC-44	U1	FPGA	Actel
AD7524JR	SOIC-16	U6	DAC	Analog Devices
TLC272ACD	SOIC-8	U2, U3	Op-Amp	Texas Instruments
ZR431F	SOT23-3	U5	Shunt regulator	Zetex
SN74HC4066	SOIC-14	U4	Analog switch	Texas Instruments
BC848B	SOT23-3	Q1, Q2, Q3, Q4	BJT npn	Motorola
BAV99	SOT23-3	D3	Dual diode	Philips
BAS16	SOT23-3	D1, D2	Single diode	Philips

Tableau A.1 : Liste des circuits intégrés et des semiconducteurs du NLB

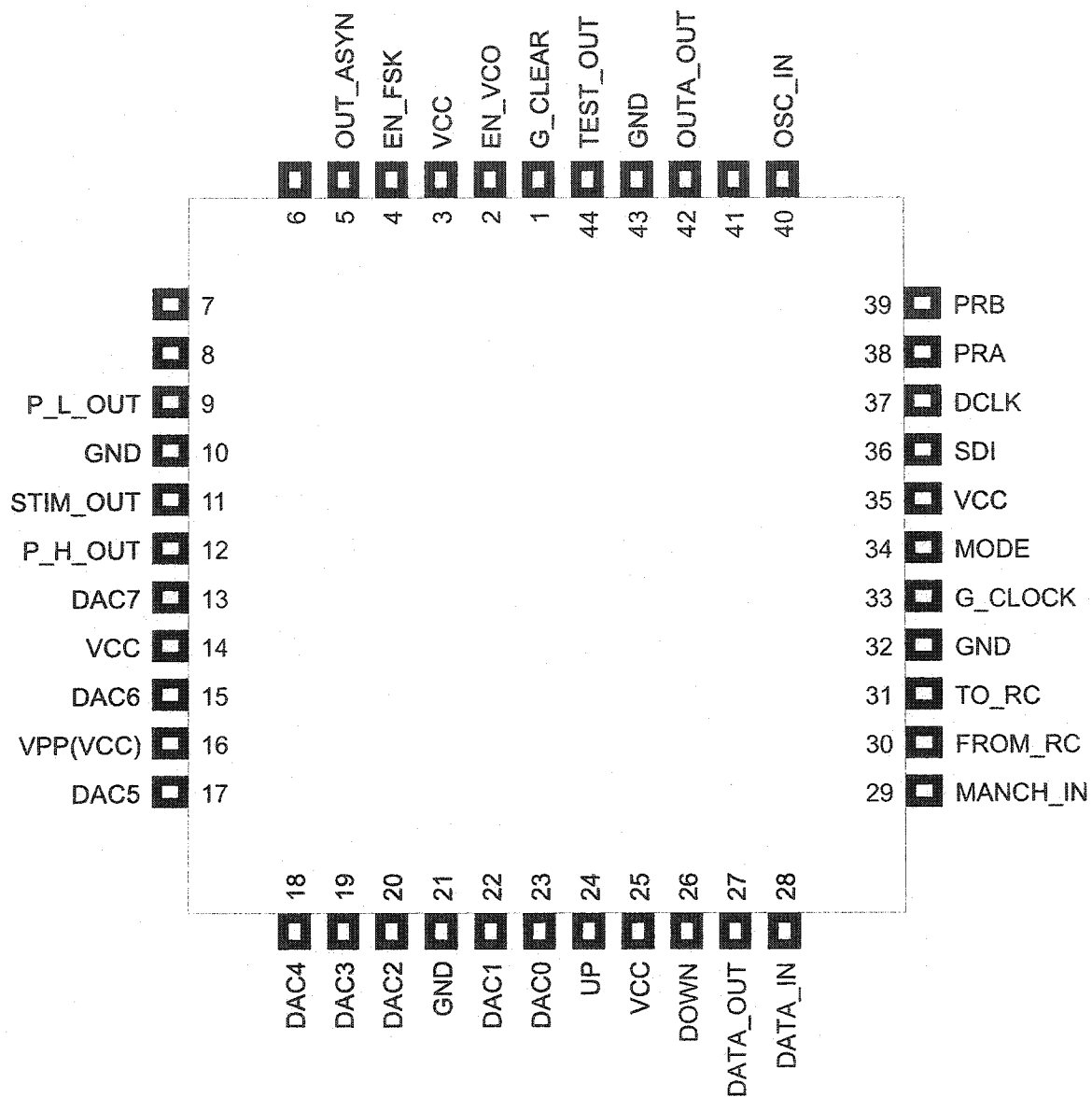


Figure A.3 : Répartition des broches pour le FPGA du NLB

File a detection d'entete (FILE)

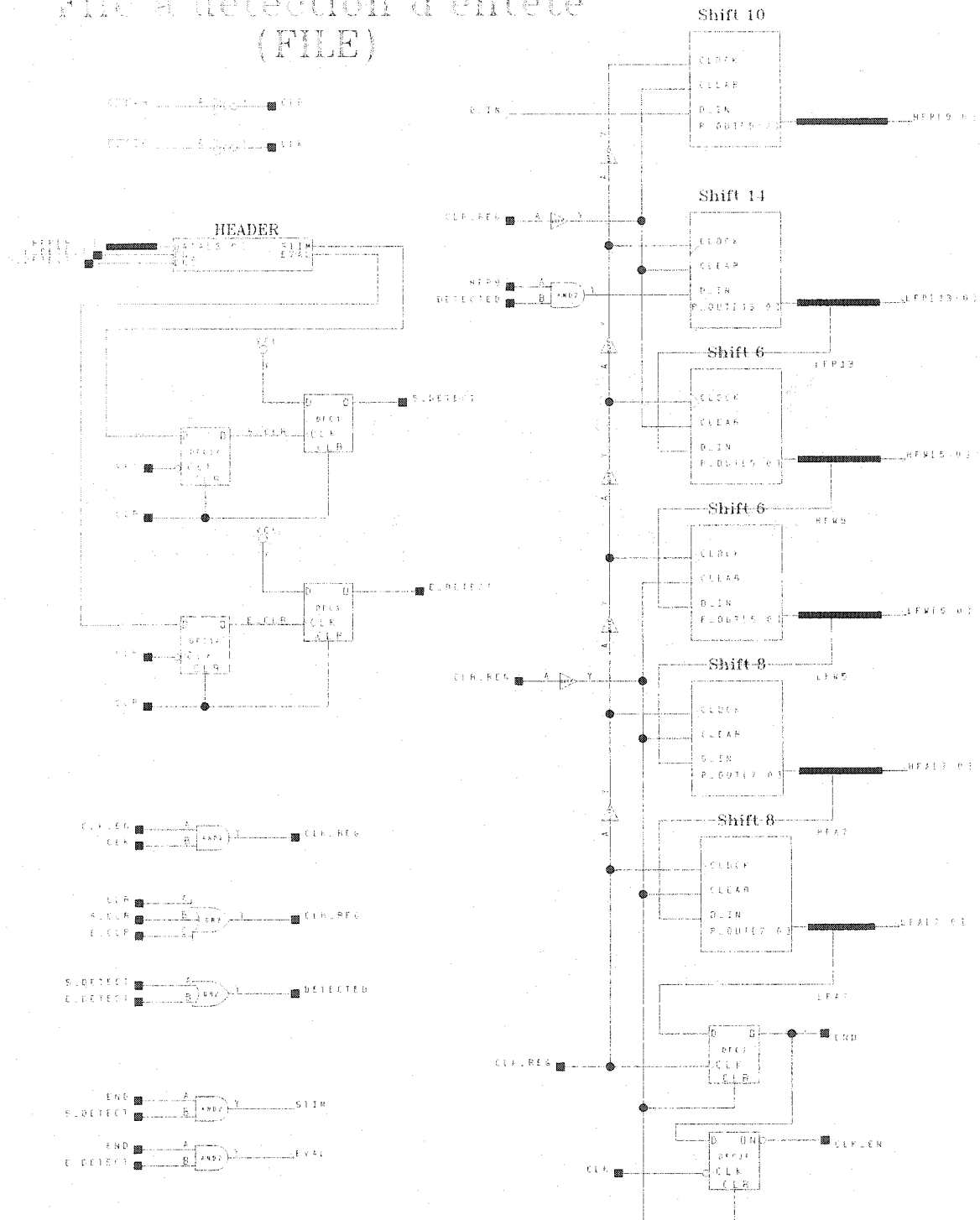


Figure A.5 : Schéma logique de la file à détection d'entête pour le FPGA du NLB

HFWgen / Square wave

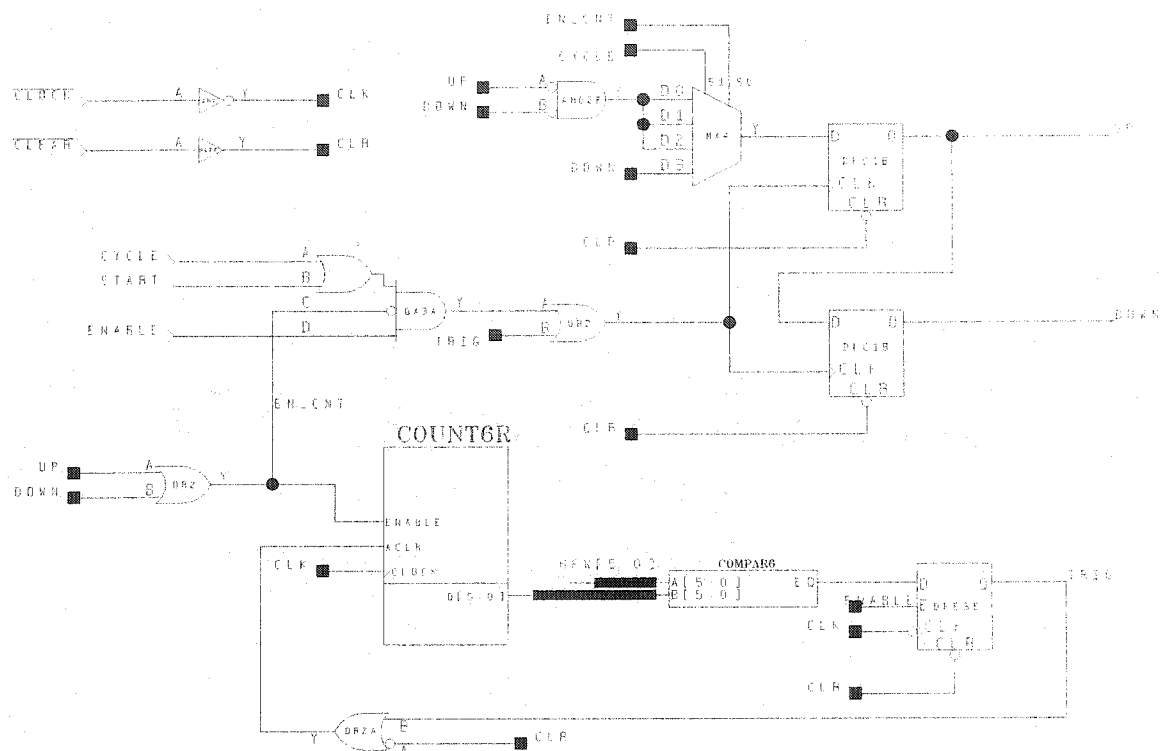


Figure A.10 : Schéma logique du contrôleur de largeur d'impulsions haute fréquence pour le FPGA du NLB

SELECT

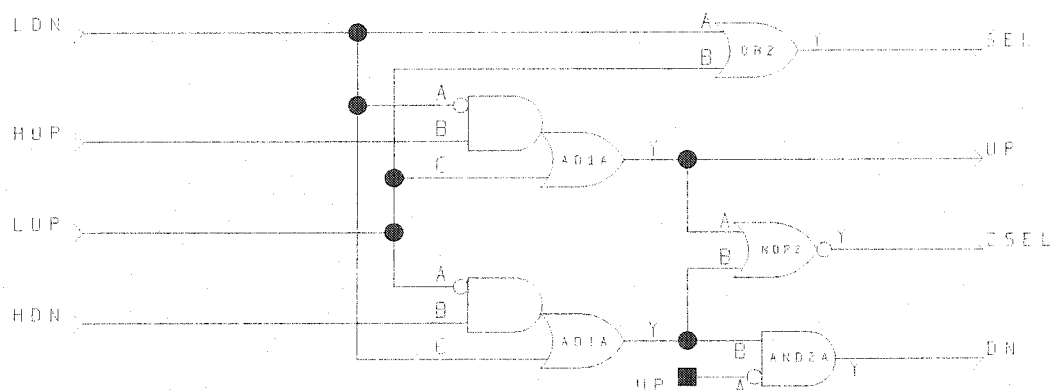


Figure A.11 : Schéma logique du sélecteur d'impulsions haut ou basse fréquence pour le FPGA du NLB

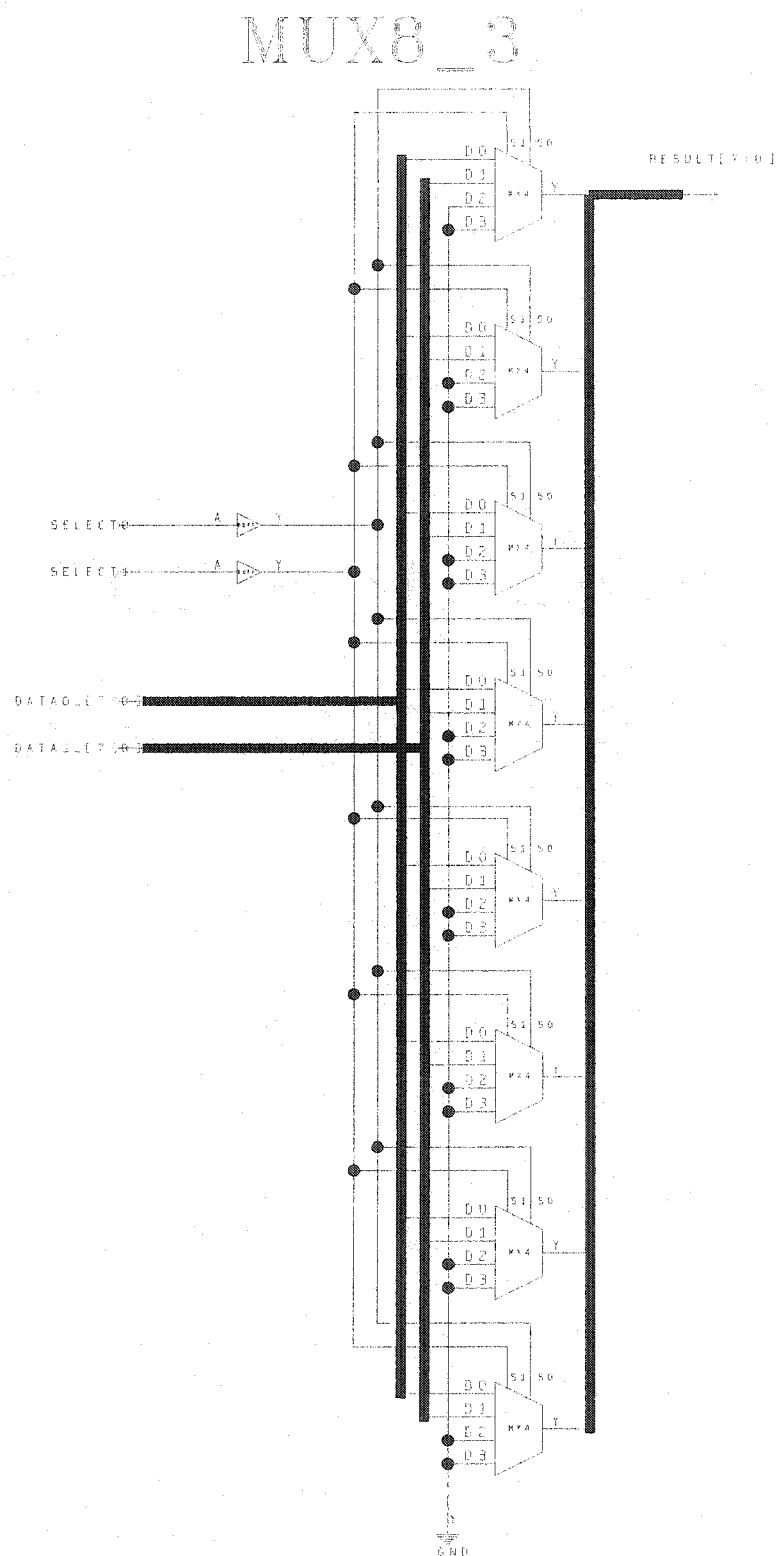


Figure A.12 : Schéma logique du multiplexeur de sélection de l'amplitude de stimulation pour le FPGA du NLB

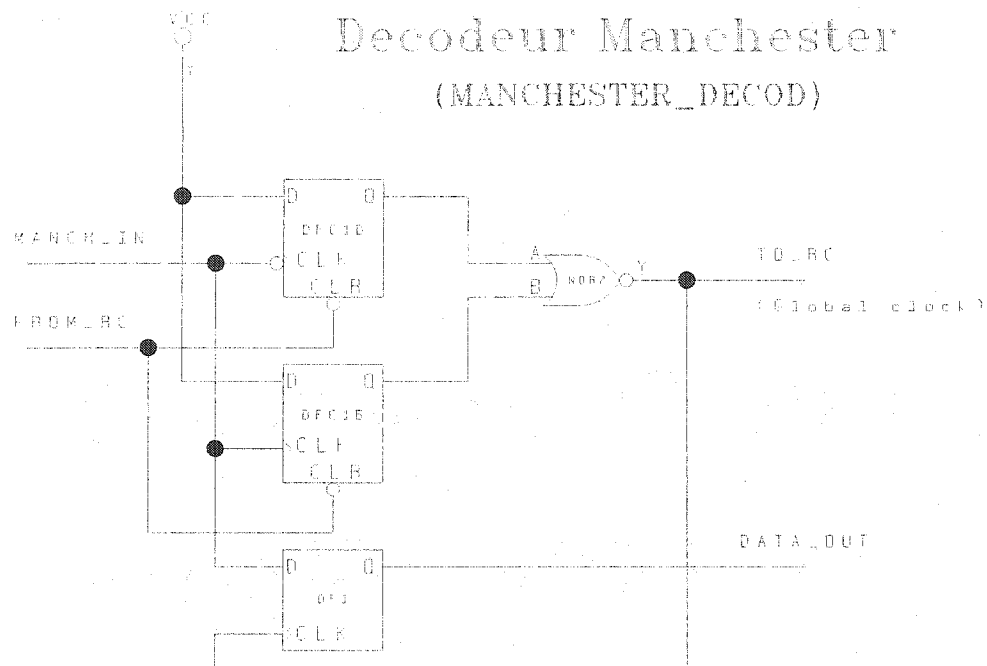


Figure A.13 : Schéma logique du décodeur Manchester pour le FPGA du NLB

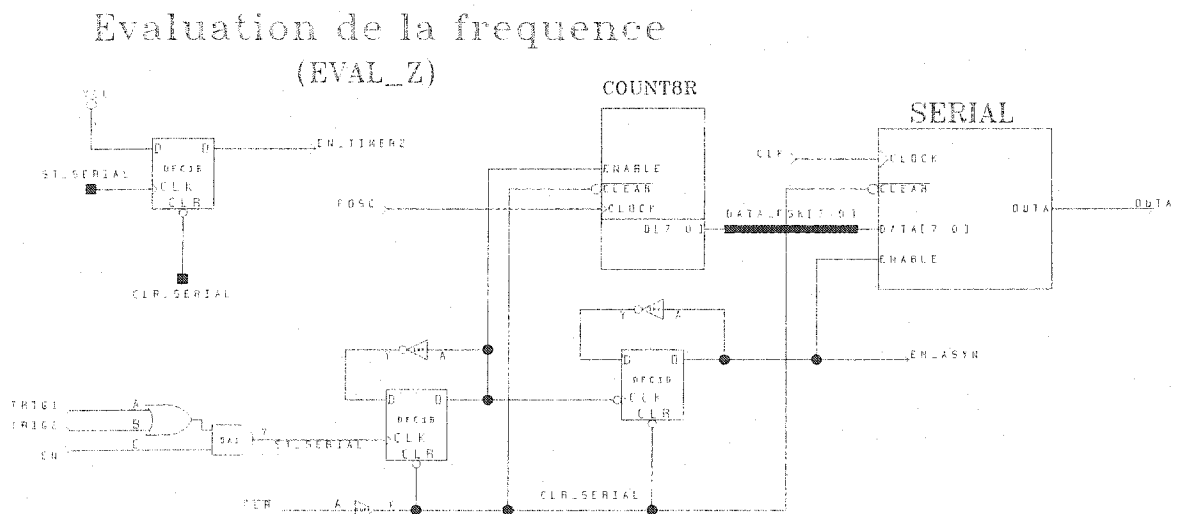


Figure A.14 : Schéma logique du compteur de fréquence pour le FPGA du NLB

SERIAL

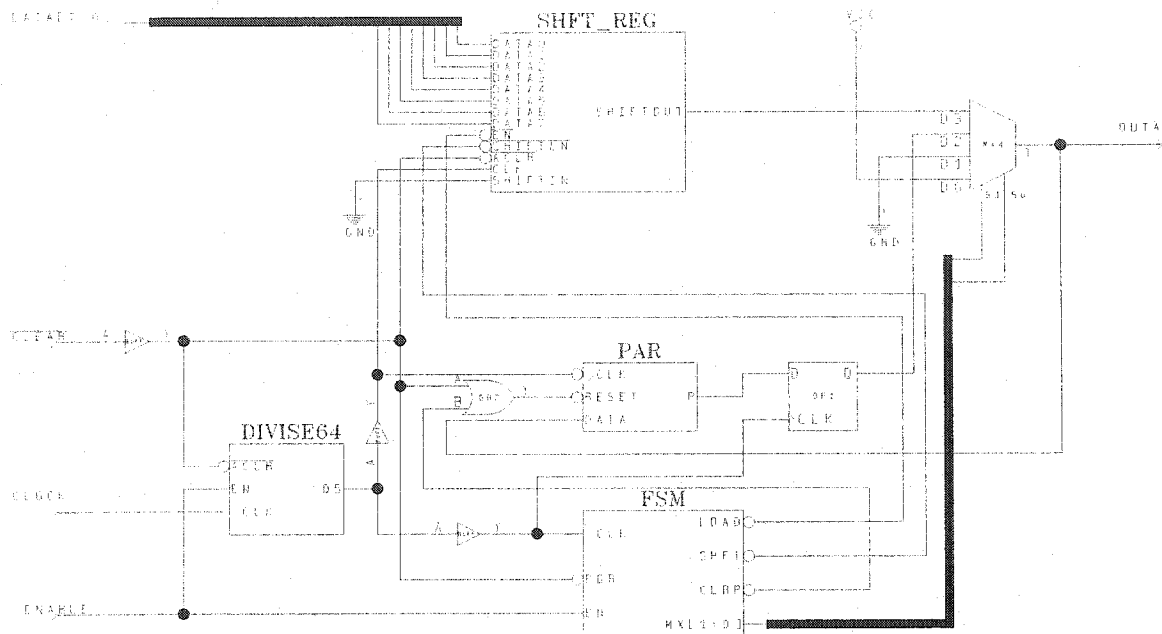


Figure A.15 : Schéma logique du convertisseur parallèle à série asynchrone pour le FPGA du NLB

```

-----
--  Nom du fichier : Header.asm --
--  Date           : 17 Fev 1999 --
--  Version        : 1.0 --
--  Auteur         : E. Schneider --
-----
--  Fichiers requis : Librairie IEEE.std_logic_1164 --
-----
--  Notes: Description VHDL de l'entité HEADER pour le FPGA du NLB --
--         Détecte le type d'entête envoyée par le contrôleur externe-
-----

library IEEE;
use IEEE.std_logic_1164.all;
entity header is
port( data    : in std_logic_vector(9 downto 0);
      en1     : in std_logic;
      en2     : in std_logic;
      stim    : out std_logic;
      eval    : out std_logic);
end header;

library IEEE;
use IEEE.std_logic_1164.all;
architecture behavior of header is
begin
  process (data,en1,en2)
  begin
    if (en1 = '0') and (en2 = '0') then
      case data is
        when "1001000111" =>
          stim <= '1';
          eval <= '0';
        when "0111111011" =>
          stim <= '0';
          eval <= '1';
        when others =>
          stim <= '0';
          eval <= '0';
        end case;
      else
        stim <= '0';
        eval <= '0';
      end if;
    end process;
  end behavior;

```

```

-----
--  Nom du fichier : Par.vhd                                --
--  Date           : 10 Janv 1999                            --
--  Version        : 1.0                                     --
--  Auteur         : E. Schneider                           --
-----
--  Fichiers requis : Librairie IEEE.std_logic_1164         --
-----
--  Notes: Description VHDL de l'entité PAR pour le FPGA du NLB --
--          Calcul la parité du mot de 8 bits à retourner par le NLB --
-----

library IEEE;
use IEEE.std_logic_1164.all;

Entity par is
  port( clk   : in  std_logic;
        reset : in  std_logic;
        data  : in  std_logic;
        p     : out std_logic);
end par;

library IEEE;
use IEEE.std_logic_1164.all;

Architecture behav00 of par is

  signal parity : std_logic;

begin
main:
  process(clk,reset)
  begin
    if reset = '0' then parity <= '0';
    elsif clk'event and clk = '0' then
      if data = '1' then
        parity <= not parity;
      end if;
    end if;
  end process;
  p <= parity;
end behav00;

```

```

-----
--  Nom du fichier : Fsm.vhd                                --
--  Date           : 10 Janv 1999                            --
--  Version        : 1.0                                     --
--  Auteur         : E. Schneider                            --
-----
--  Fichiers requis : Librairie IEEE.std_logic_1164         --
-----
--  Notes: Description VHDL de l'entité FSM pour le FPGA du NLB --
--          Machine séquentielle algorithmique pour le contrôle de --
--          la conversion et de l'envoi des données par le NLB    --
-----

library IEEE;
use IEEE.std_logic_1164.all;

Entity fsm is
  port( clk      : in  std_logic;
        por      : in  std_logic;
        en       : in  std_logic;
        load     : out std_logic;
        shft     : out std_logic;
        clrp     : out std_logic;
        mx       : out std_logic_vector(1 downto 0));
end fsm;

library IEEE;
use IEEE.std_logic_1164.all;

Architecture behavior of fsm is

  type states is (waiting,Loading,shifting,parity,stop_bit);
  type num_3bits is range 0 to 7;

begin
main:
  process(clk,por)
    variable state : states;
    variable i     : num_3bits;
  begin
    if por = '0' then
      state := waiting;
      load  <= '1';
      shft  <= '1';
      clrp  <= '0';
      mx    <= "00";
    elsif clk'event and clk = '1' then
      case state is
        when waiting =>
          if en = '1' then
            state := loading;
            load  <= '0';
            shft  <= '1';
            clrp  <= '0';
            mx    <= "01";
          end if;
        -- Loading
        -- Shifting
        -- Parity
        -- Stop bit
      end case;
    end if;
  end process;
end behavior;

```

```

else
    state := waiting;
    load  <= '1';
    shft  <= '1';
    clrp  <= '0';
    mx    <= "00";
end if;
when loading =>
    state := shifting;
    load  <= '1';
    shft  <= '0';
    clrp  <= '1';
    mx    <= "11";
    i     := 7;
when shifting =>
    if i = 0 then
        state := parity;
        load  <= '1';
        shft  <= '1';
        clrp  <= '1';
        mx    <= "10";
    else
        state := shifting;
        load  <= '1';
        shft  <= '0';
        clrp  <= '1';
        mx    <= "11";
        i     := i - 1;
    end if;
when parity =>
    state := stop_bit;
    load  <= '1';
    shft  <= '1';
    clrp  <= '0';
    mx    <= "00";
when stop_bit =>
    state := waiting;
    load  <= '1';
    shft  <= '1';
    clrp  <= '0';
    mx    <= "00";
end case;
end if;
end process;
end behavior;

```

ANNEXE B

SCHÉMAS ET CODES DU NSP

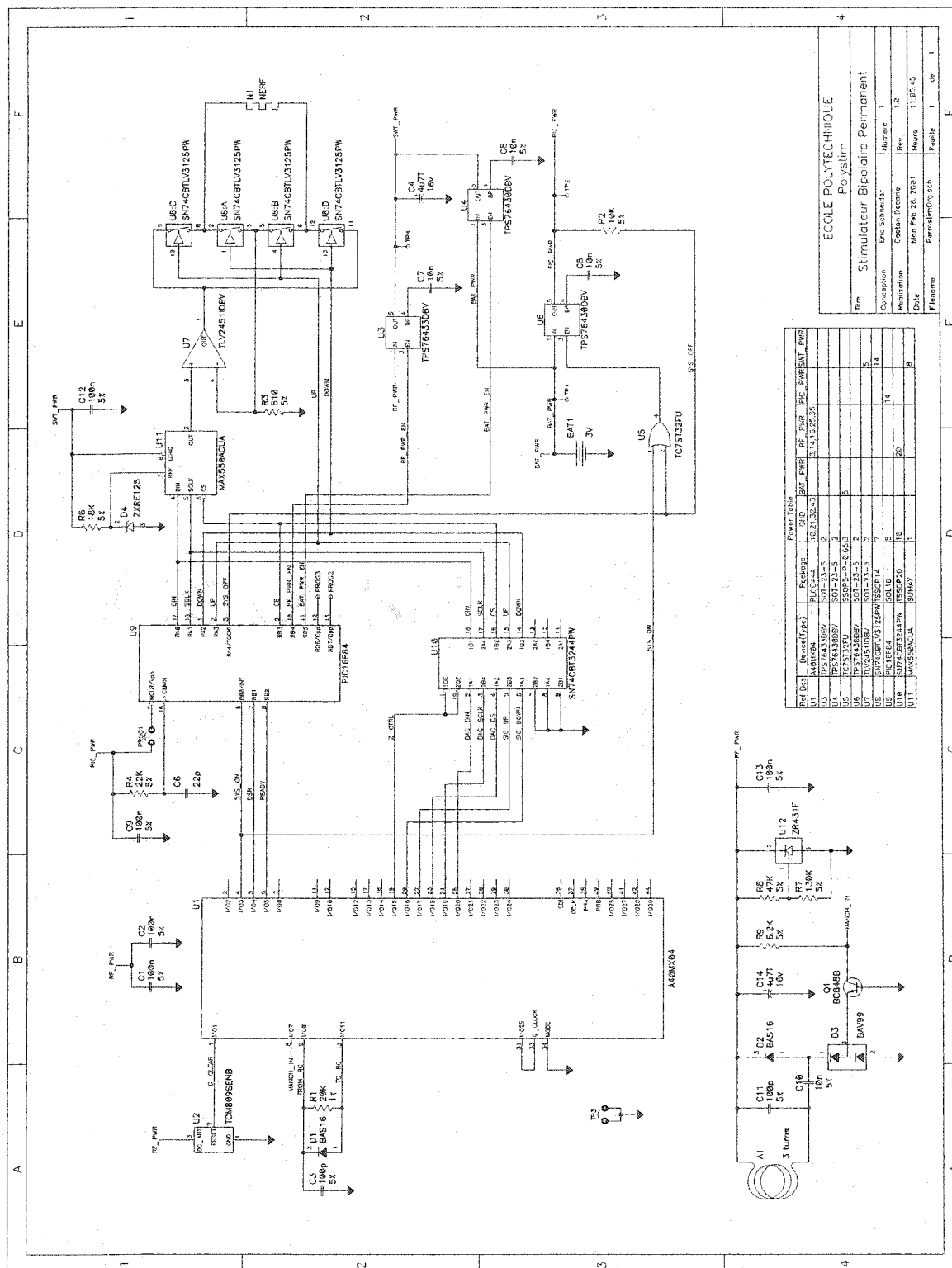


Figure B.1 : Schéma électrique initial du NSP

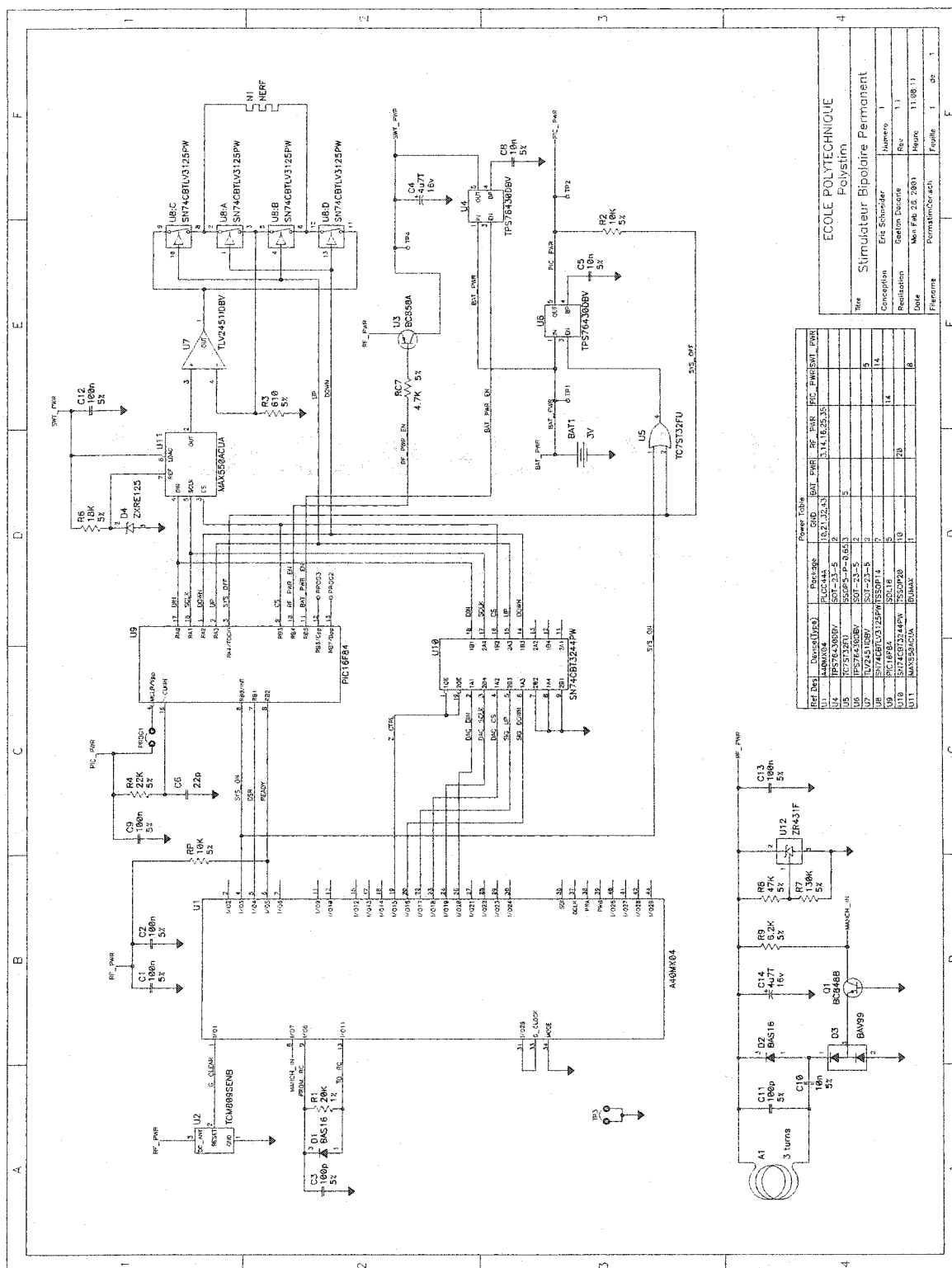


Figure B.2 : Schéma électrique corrigé du NSP

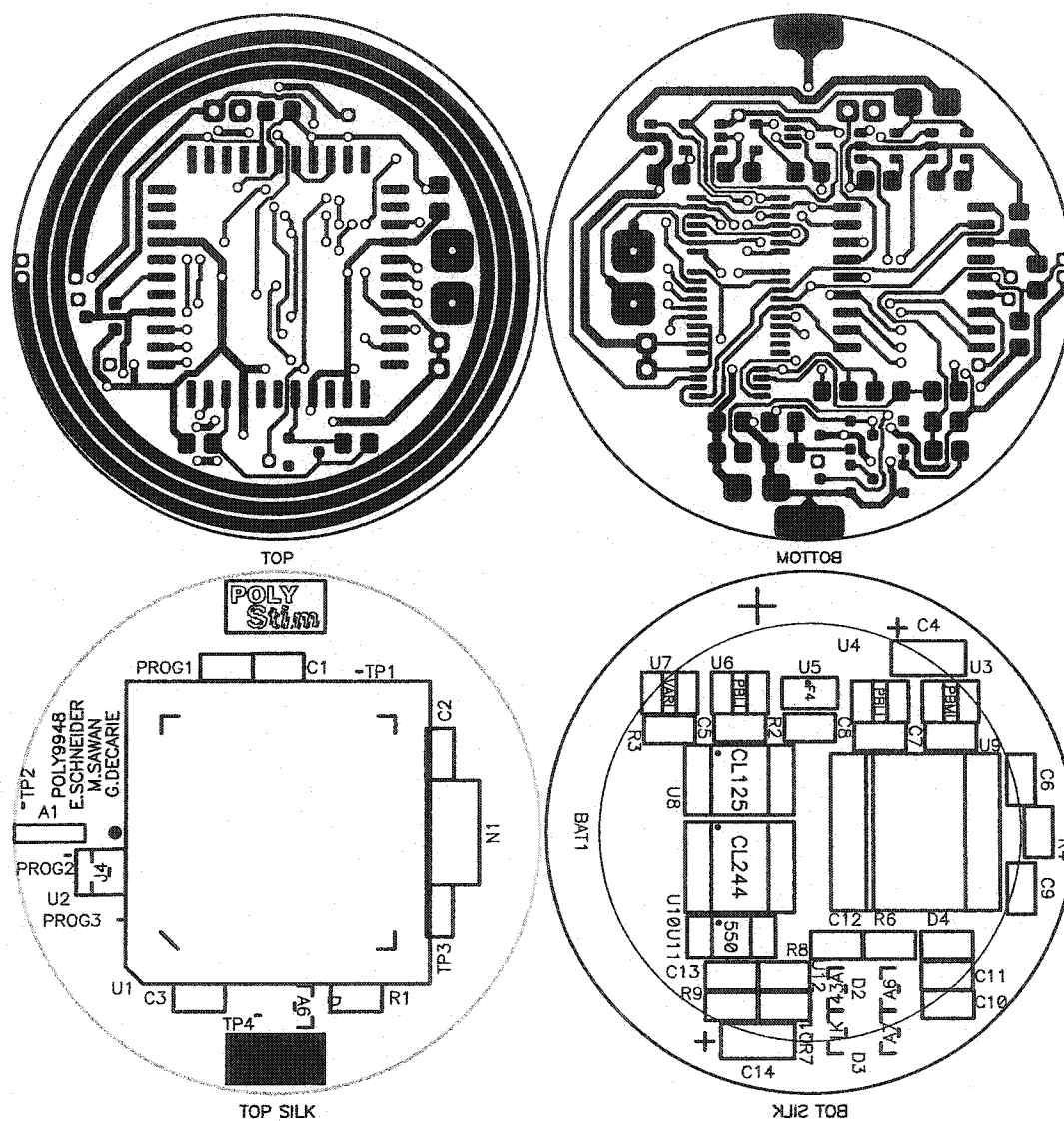


Figure B.3 : Dessin des masques initial à l'échelle 2/1 du NSP

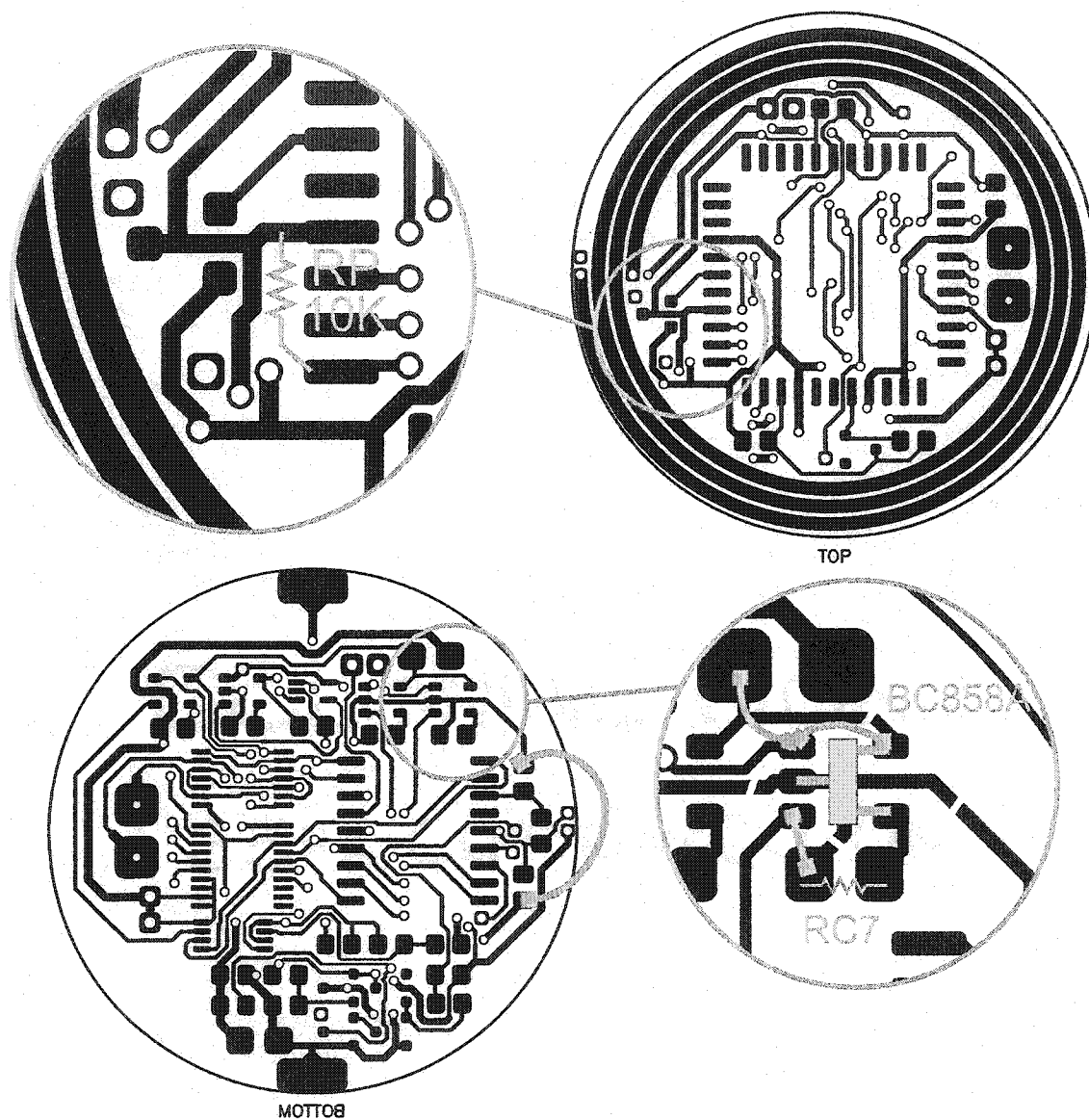


Figure B.4 : Modifications apportées au dessin des masques du NSP

Numéro de référence	Boîtier	Identificateur	Description	Fabricant
A40MX04	PLCC-44	U1	FPGA	Actel
PIC16F84-04/SO	SOIC-18	U9	PICmicro	Microchip
MAX550ACUA	uMAX-8	U11	Serial DAC	Maxim
SN74CBTLV3245A	PW-20	U10	Bus switch	Texas Instruments
SN74CBTLV3125	PW-14	U8	Bus switch	Texas Instruments
TC7ST32FU	USV-5	U5	Single OR gate	Toshiba
TCM809SENB	SOT23B-3	U2	POR/BOR	Telcom Semi
TPS76430DBV	DBV-5	U4, U6	Regulator	Texas Instruments
TPS76433DBV	DBV-5	U3	Regulator	Texas Instruments
TLV2451IDBV	DBV-5	U7	Op-Amp	Texas Instruments
ZR431F	SOT23-3	U12	Shunt regulator	Zetex
ZXRE125F	SOT23-3	D4	Tension Ref	Zetex
BAS16	SOT23-3	D1, D2	Single Diode	Philips
BAV99	SOT23-3	D3	Dual diode	Philips
BC848B	SOT23-3	Q1	BJT npn	Motorola
BC858A	SOT23-3	U3	BJT pnp	Motorola

Tableau B.1 : Liste des circuits intégrés et des semiconducteurs du NSP

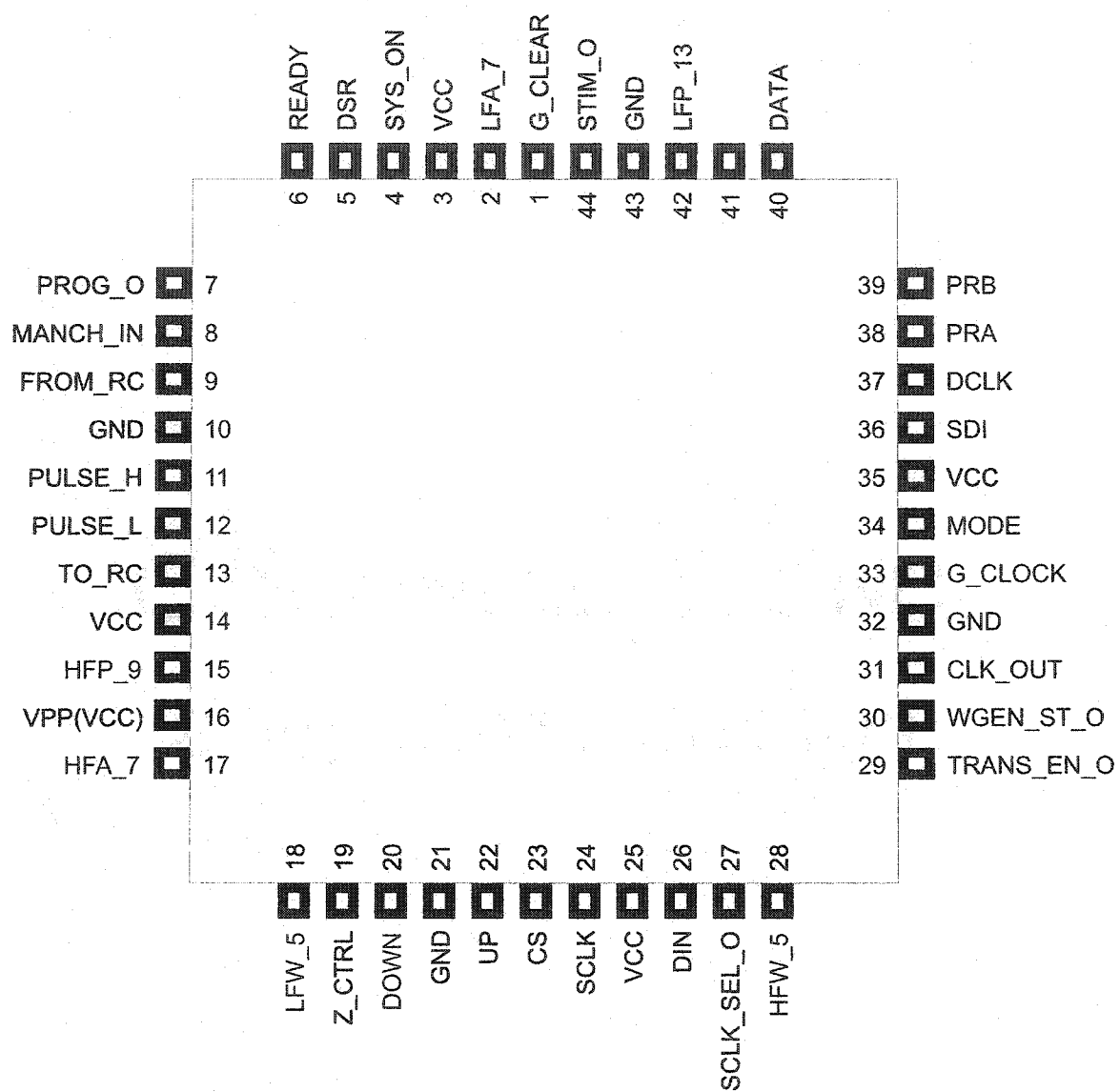


Figure B.5 : Répartition des broches pour le FPGA du NSP

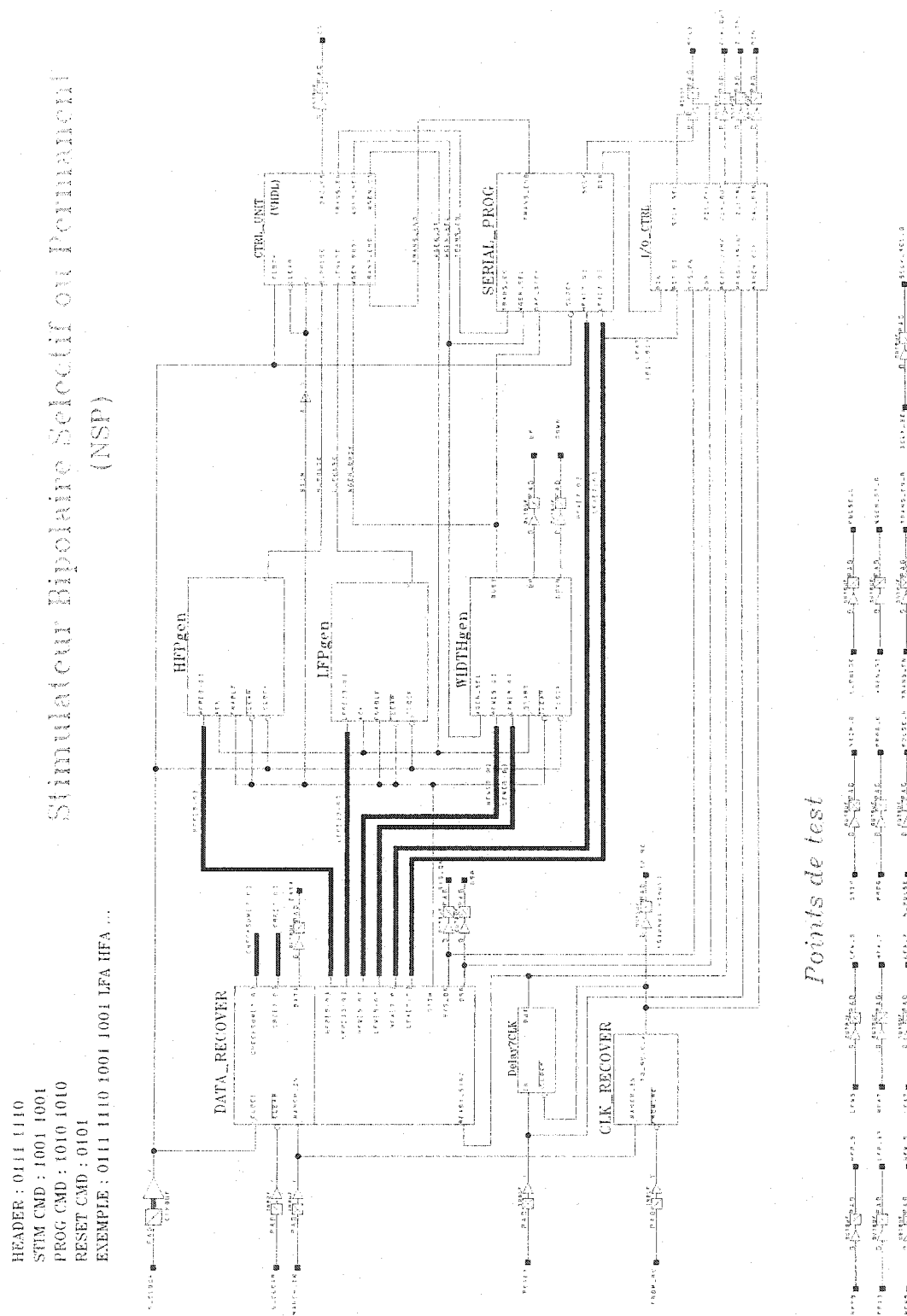


Figure B.6 : Schéma logique de niveau hiérarchique supérieur pour le FPGA du NSP

HEADER : 0111 1110
 STIM CMD : 1001 1001
 PROG CMD : 1010 1010
 RESET CMD : 0101
 EXEMPLE : 0111 1110 1001 1001 LFA HFA ...

File à detection d'entete (DATA_RECOVER)

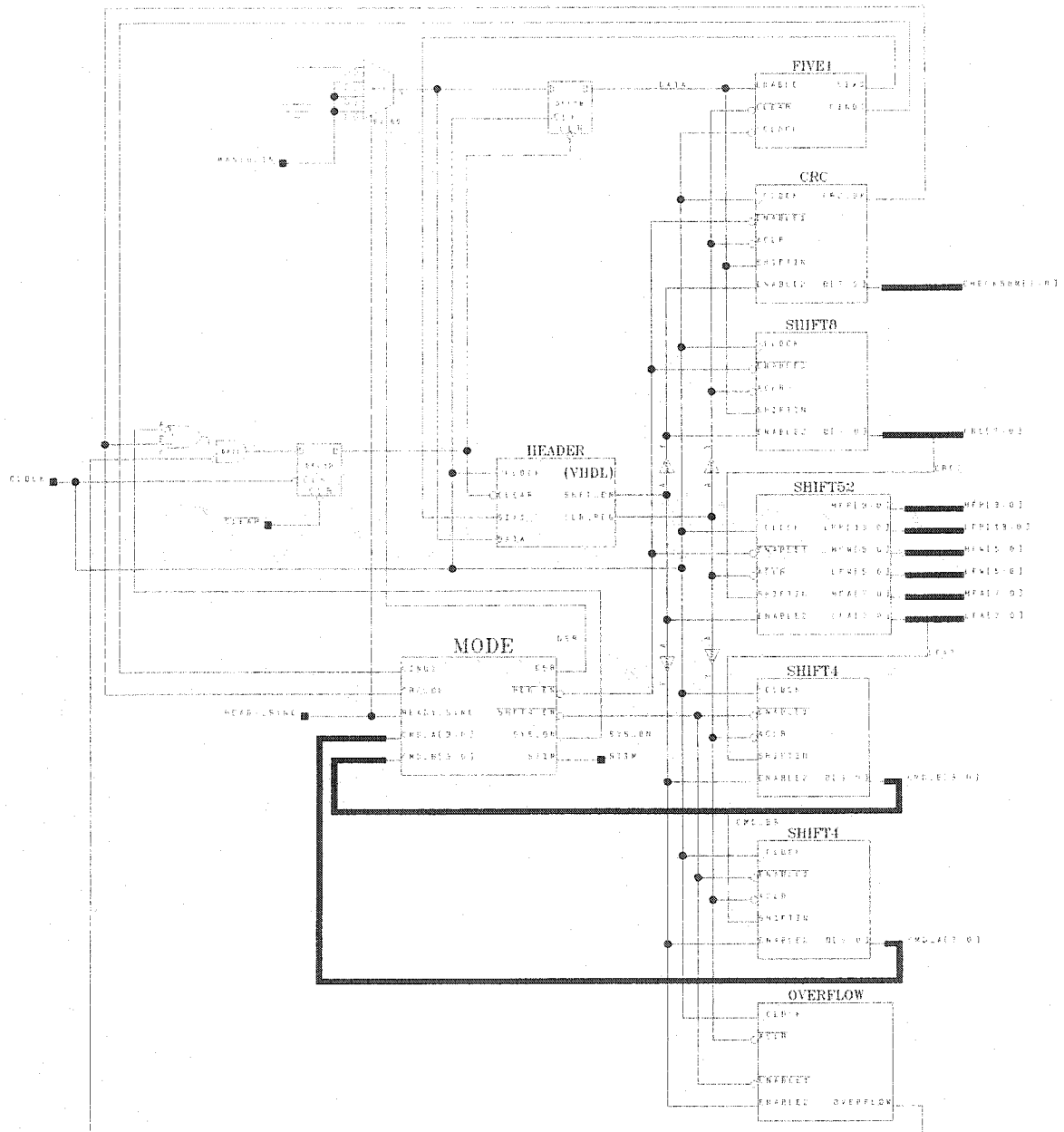


Figure B.7 : Schéma logique de la file à détection d'entête pour le FPGA du NSP

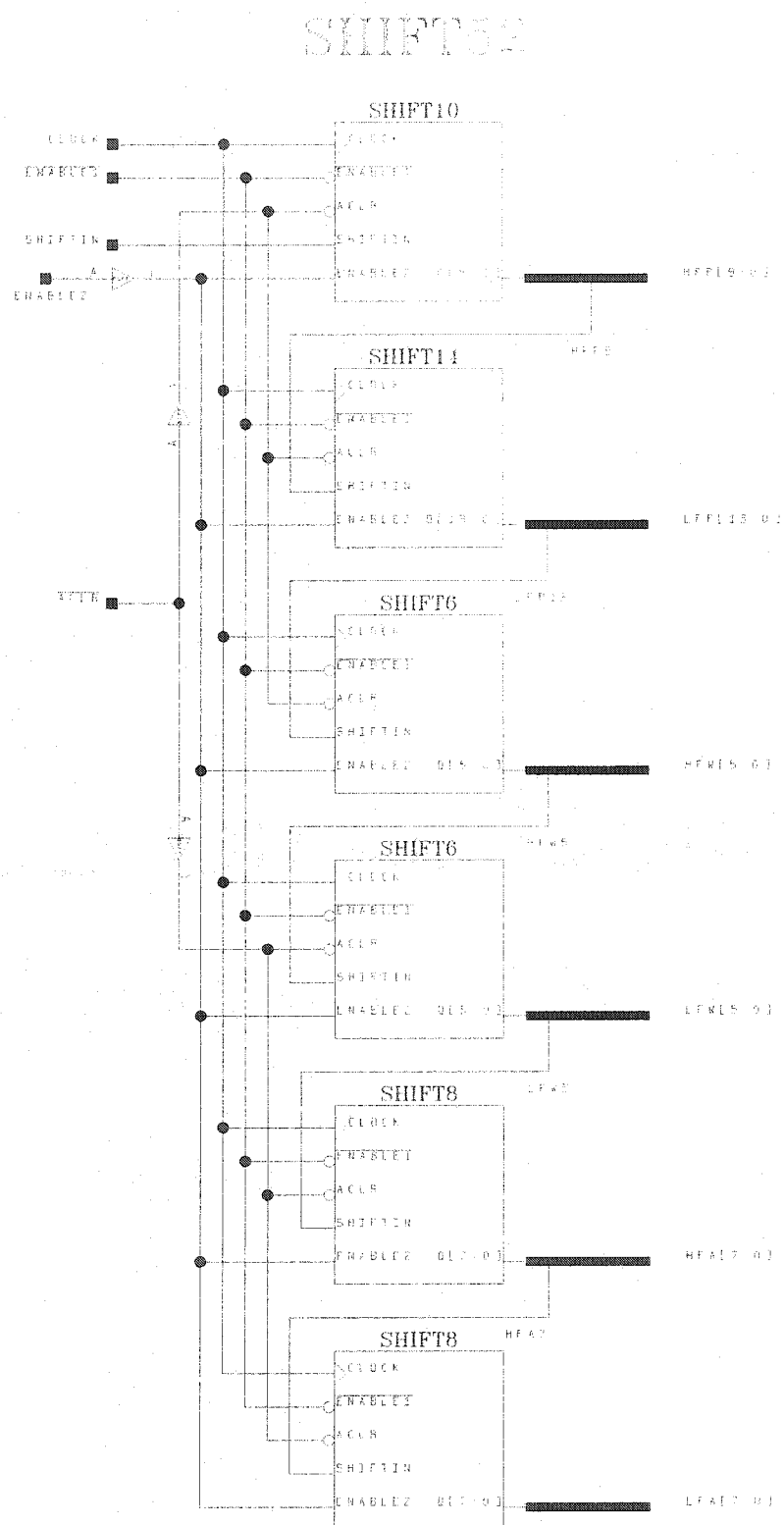


Figure B.8 : Schéma logique du registre à décalage des paramètres de stimulation pour le FPGA du NSP

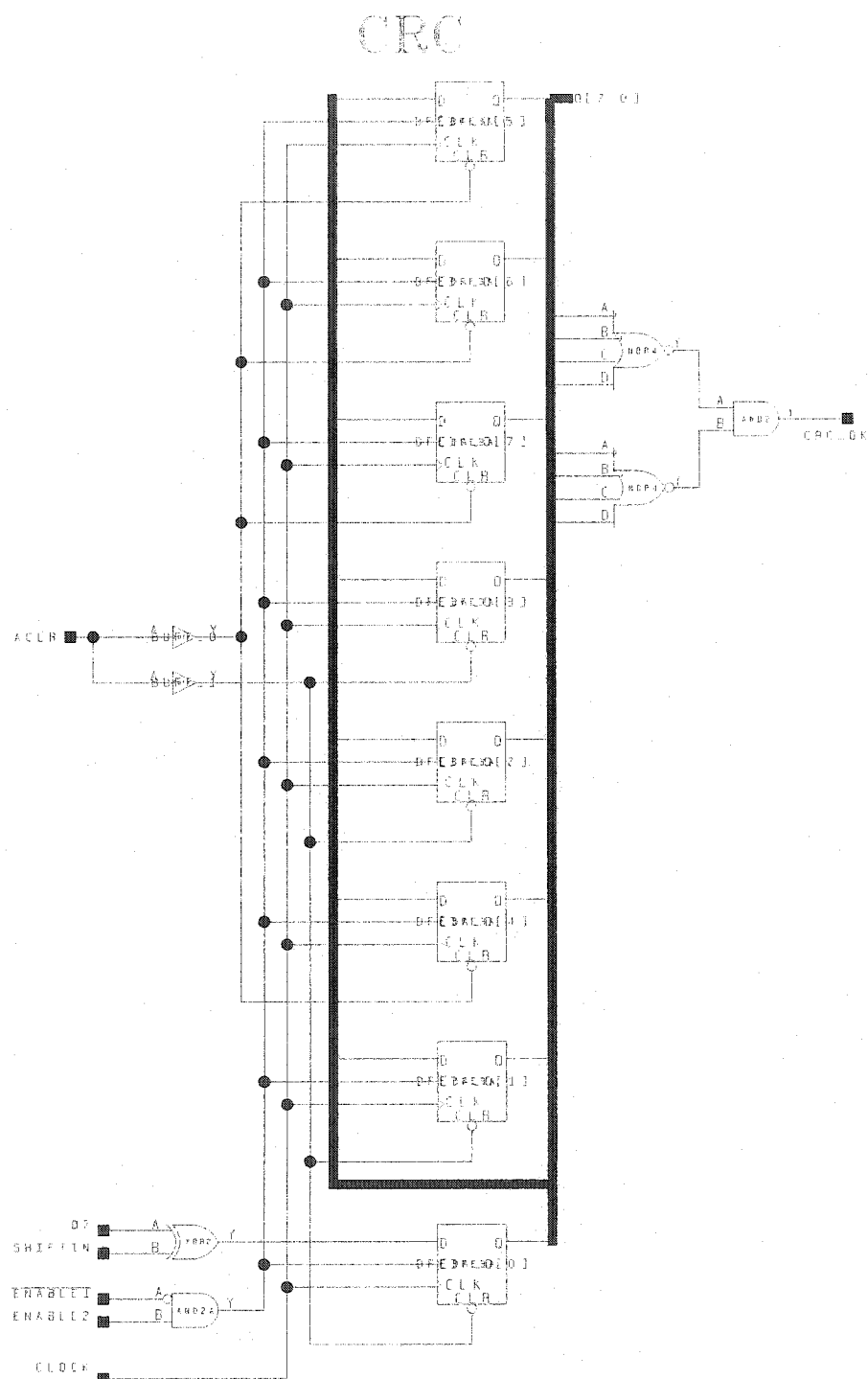


Figure B.10 : Schéma logique du calculateur de CRC pour le FPGA du NSP

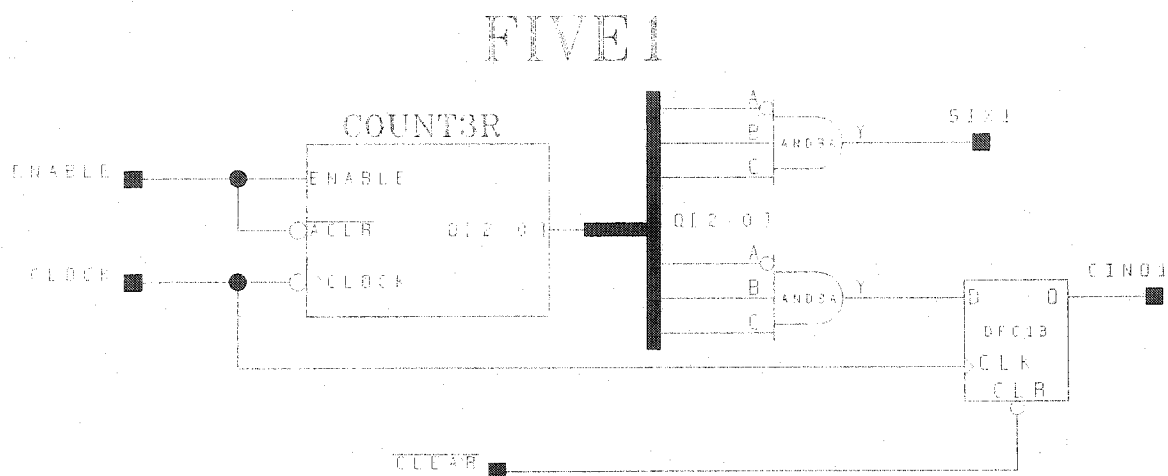


Figure B.12 : Schéma logique du détecteur de cinq et de six "1" consécutifs dans une trame pour le FPGA du NSP

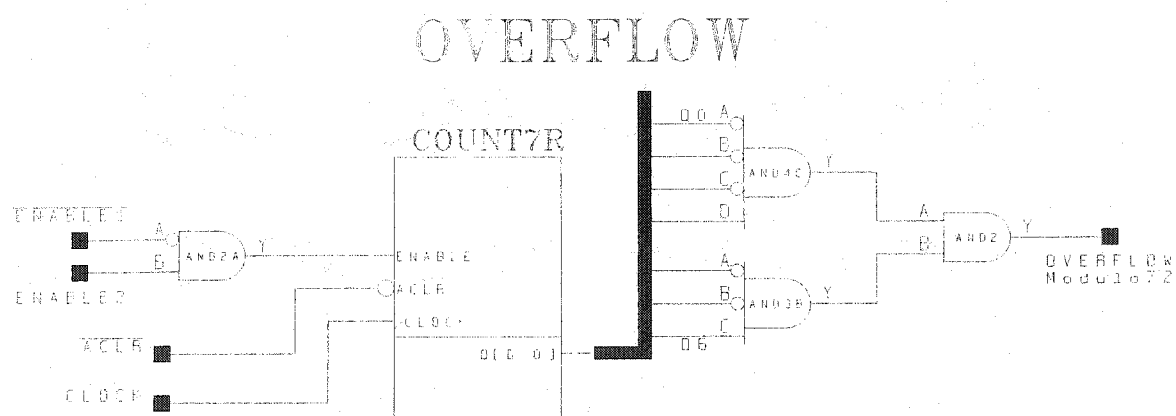


Figure B.13 : Schéma logique du détecteur de dépassement de longueur de trame pour le FPGA du NSP

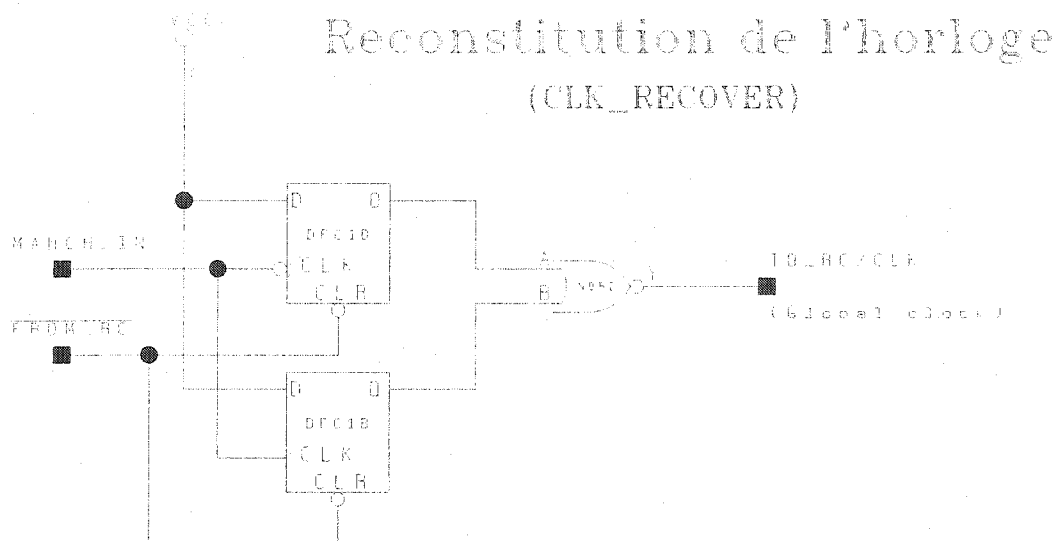


Figure B.14 : Schéma logique du décodeur Manchester pour le FPGA du NSP

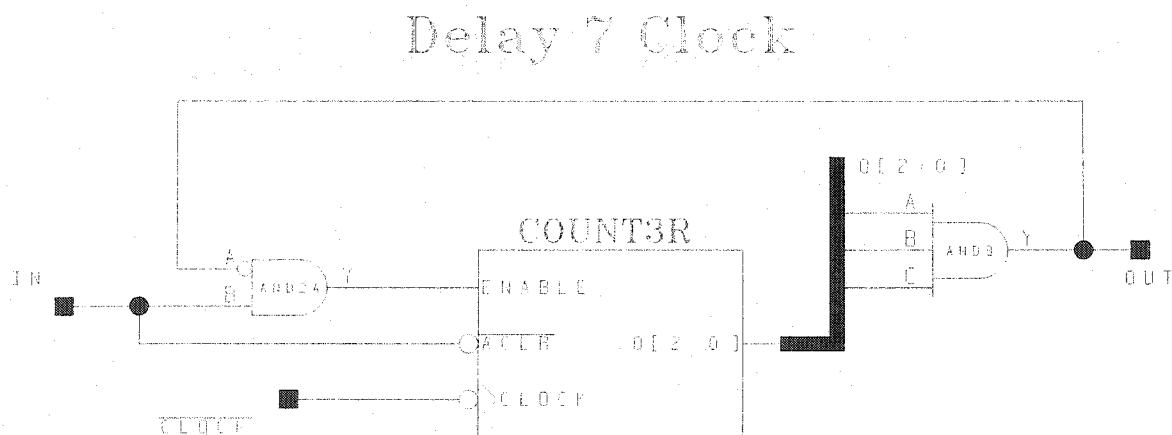


Figure B.15 : Schéma logique du temporisateur de synchronisation pour le FPGA du NSP

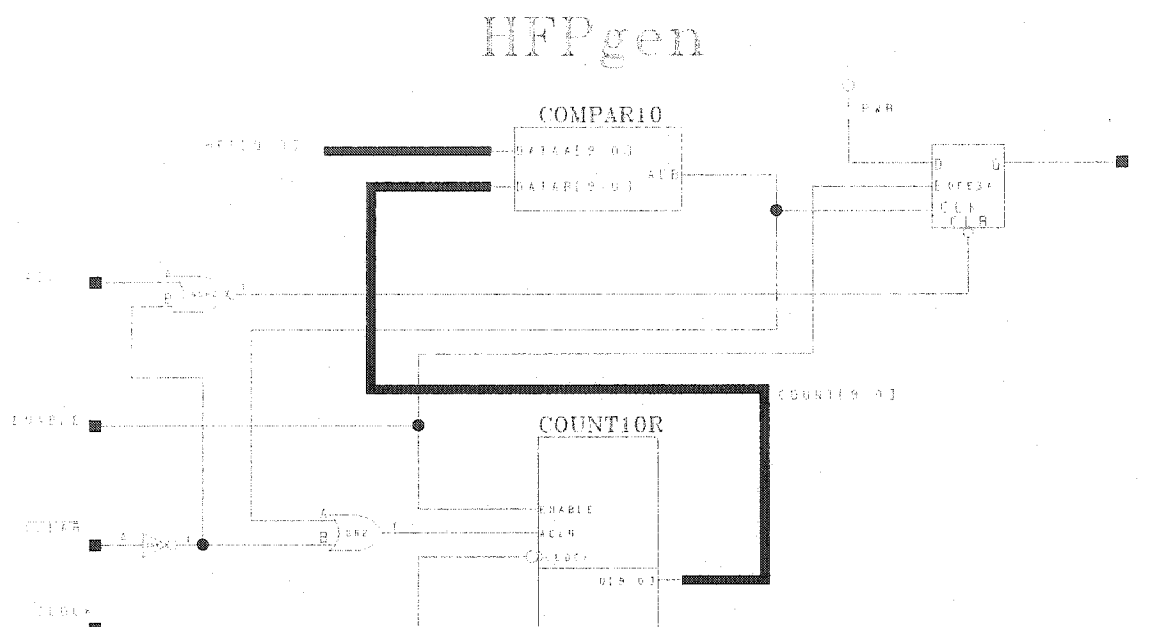


Figure B.16 : Schéma logique du générateur d'impulsions haute fréquence pour le FPGA du NSP

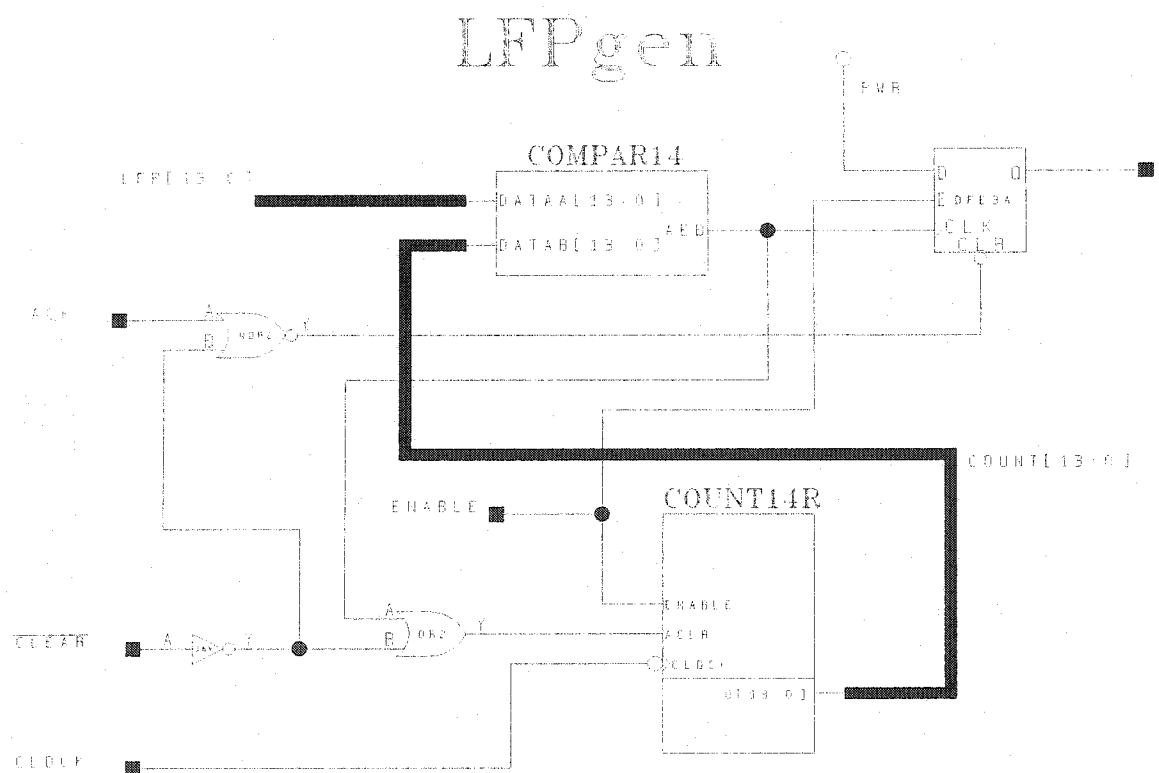


Figure B.17 : Schéma logique du générateur d'impulsions basse fréquence pour le FPGA du NSP

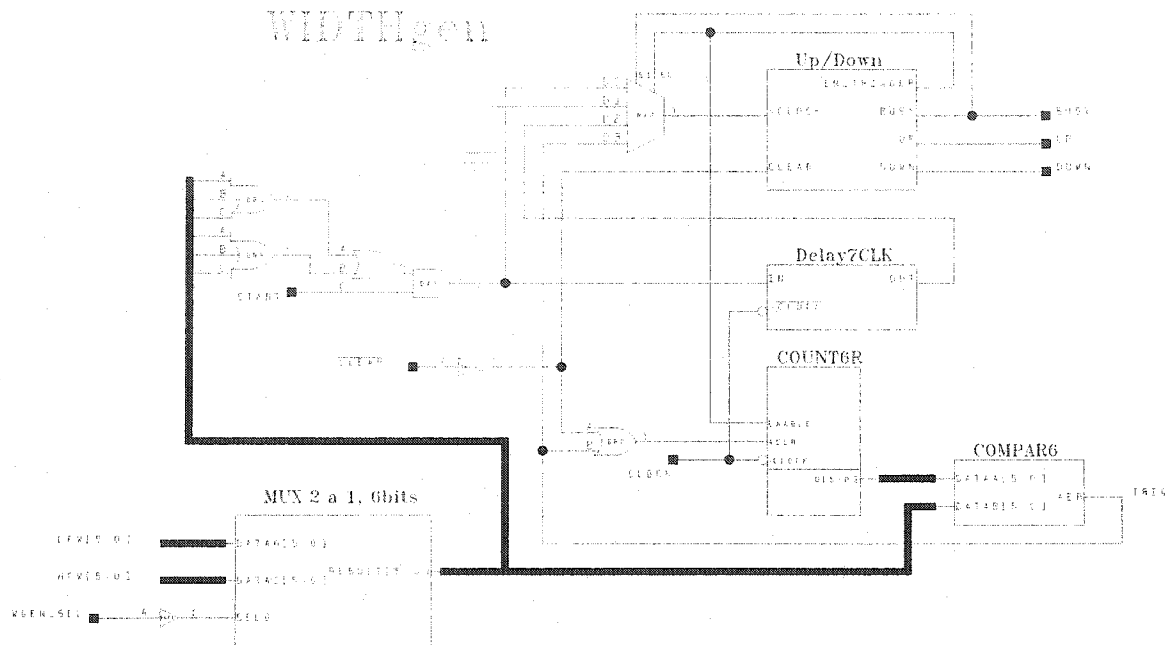


Figure B.18 : Schéma logique du contrôleur de largeur d'impulsions pour le FPGA du NSP

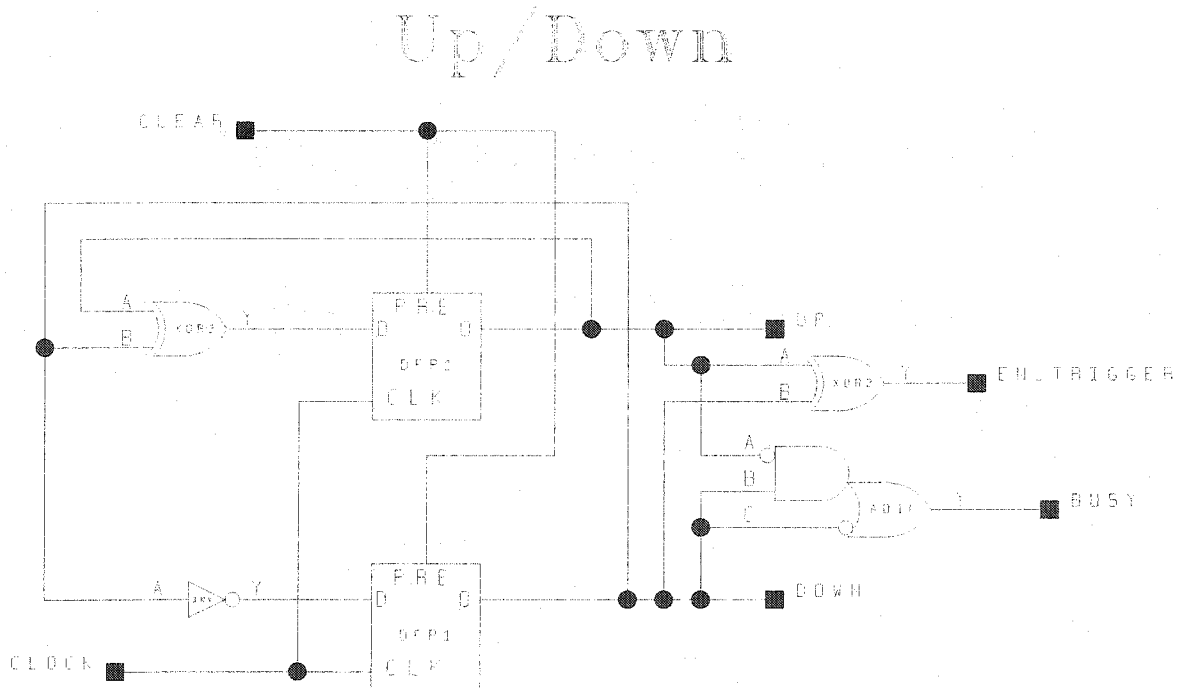


Figure B.19 : Schéma logique du contrôleur de commutation du sens du courant de stimulation pour le FPGA du NSP

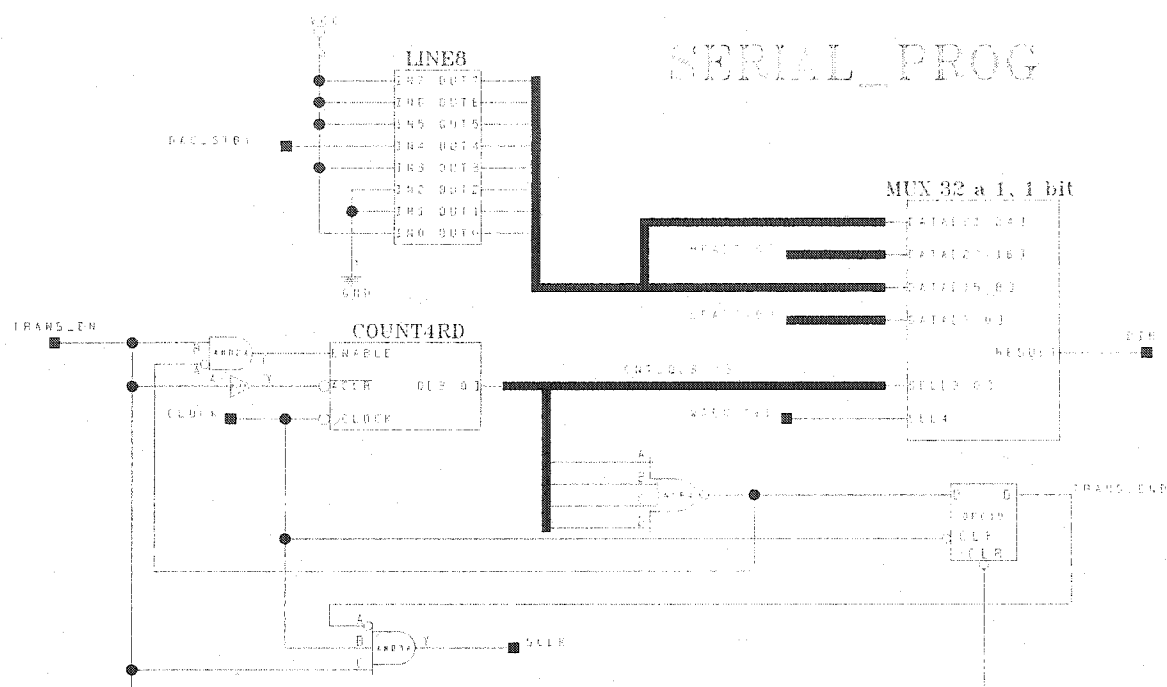


Figure B.20 : Schéma logique du générateur de commandes pour DAC sériel pour le FPGA du NSP

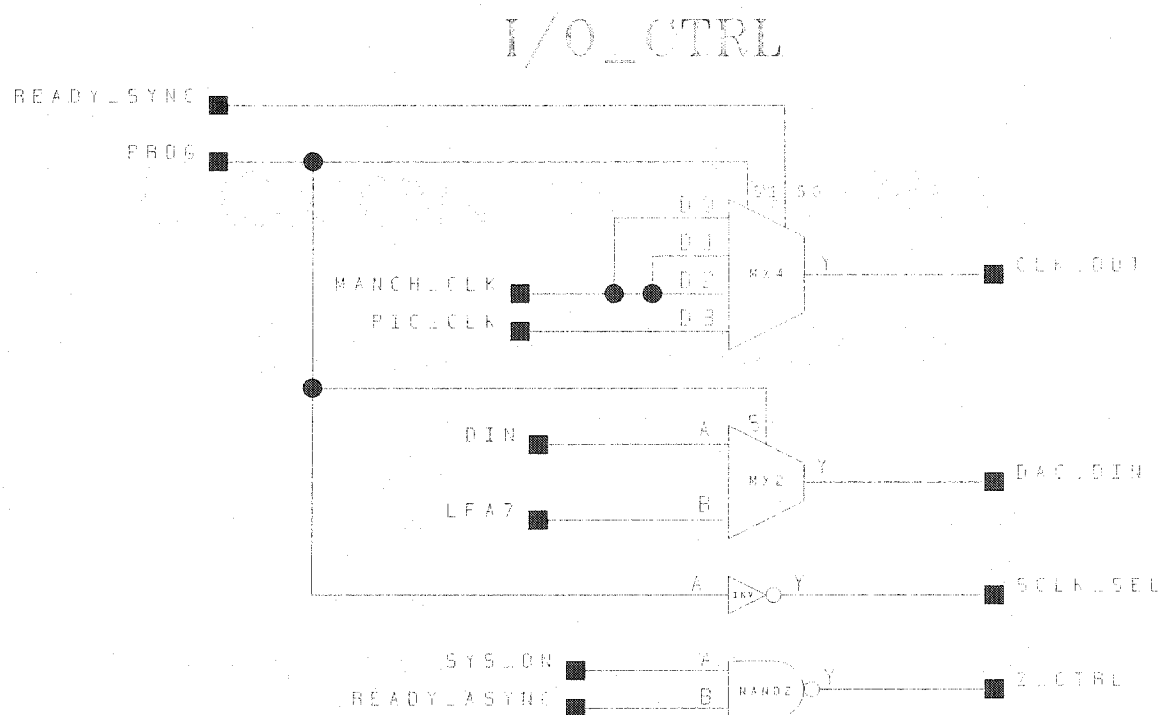


Figure B.21 : Schéma logique du contrôleur des Entrées/Sorties pour le FPGA du NSP

```

-----
--  Nom du fichier : Header.vhd                                --
--  Date           : 17 Dec 1999                                --
--  Version        : 1.0                                         --
--  Auteur         : E. Schneider                                --
-----
--  Fichiers requis : Librairie IEEE.std_logic_1164            --
-----
--  Notes: Description VHDL de la machine à états "HEADER" utilisée --
--         dans la file à détection d'entête dans le FPGA du NSP  --
-----

```

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

```

```

ENTITY HEADER IS
    PORT (CLOCK,CLEAR,DATA,SIX1 : IN std_logic;
          SHFT_EN,CLR_REG : BUFFER std_logic);
END;

```

```

ARCHITECTURE BEHAVIOR OF HEADER IS
    TYPE ETATS IS (ATTENTE,ETAT1,ETAT2,ETAT3,ETAT4,ETAT5,ETAT6);
BEGIN
    PROCESS (CLOCK,CLEAR)
        VARIABLE ETAT : ETATS;
    BEGIN
        IF ( CLEAR='0' ) THEN
            ETAT := ATTENTE;
            SHFT_EN <= '0';
            CLR_REG <= '0';
        ELSIF CLOCK='1' AND CLOCK'event THEN
            CASE ETAT IS
                WHEN ATTENTE =>
                    IF DATA='0' AND SIX1='1' THEN
                        ETAT := ETAT1;
                        SHFT_EN <= SHFT_EN;
                        CLR_REG <= CLR_REG;
                    ELSE
                        ETAT := ATTENTE;
                        SHFT_EN <= SHFT_EN;
                        CLR_REG <= CLR_REG;
                    END IF;
                WHEN ETAT1 =>
                    IF DATA='1' THEN
                        ETAT := ETAT2;
                        SHFT_EN <= '1';
                        CLR_REG <= '1';
                    ELSE
                        ETAT := ETAT3;
                        SHFT_EN <= SHFT_EN;
                        CLR_REG <= CLR_REG;
                    END IF;
                WHEN ETAT2 =>
                    ETAT := ATTENTE;
            END CASE;
        END IF;
    END PROCESS;

```

```

    SHFT_EN <= SHFT_EN;
    CLR_REG <= CLR_REG;
WHEN ETAT3 =>
    IF DATA='1' THEN
        ETAT := ETAT4;
        SHFT_EN <= SHFT_EN;
        CLR_REG <= CLR_REG;
    ELSE
        ETAT := ATTENTE;
        SHFT_EN <= SHFT_EN;
        CLR_REG <= CLR_REG;
    END IF;
WHEN ETAT4 =>
    IF DATA='0' THEN
        ETAT := ETAT5;
        SHFT_EN <= SHFT_EN;
        CLR_REG <= CLR_REG;
    ELSE
        ETAT := ATTENTE;
        SHFT_EN <= SHFT_EN;
        CLR_REG <= CLR_REG;
    END IF;
WHEN ETAT5 =>
    IF DATA='1' THEN
        ETAT := ETAT6;
        SHFT_EN <= '0';
        CLR_REG <= '0';
    ELSE
        ETAT := ATTENTE;
        SHFT_EN <= SHFT_EN;
        CLR_REG <= CLR_REG;
    END IF;
WHEN ETAT6 =>
    ETAT := ATTENTE;
    SHFT_EN <= SHFT_EN;
    CLR_REG <= CLR_REG;
END CASE;
END IF;
END PROCESS;
END BEHAVIOR;

```

```

-----
--  Nom du fichier : CTRLUNIT.vhd                                --
--  Date           : 16 Dec 1999                                --
--  Version        : 1.0                                         --
--  Auteur         : E. Schneider                                --
-----
--  Fichiers requis : Librairie IEEE.std_logic_1164             --
-----
--  Notes: Description VHDL de la machine à états "CTRLUNIT" qui --
--          supervise la génération des stimuli dans le FPGA du NSP --
-----

```

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

```

```

ENTITY CTRLUNIT IS
PORT (CLOCK,CLEAR,EN,L_PULSE,H_PULSE : IN std_logic;
      WGEN_BUSY,TRANS_END : IN std_logic;
      DAC_CS,TRANS_EN : OUT std_logic;
      WGEN_SEL,WGEN_ST : BUFFER std_logic);
END;

```

```

ARCHITECTURE BEHAVIOR OF CTRLUNIT IS
TYPE STATES IS
(IDLE,STATE0,STATE1A,STATE2A,STATE1B,STATE2B,STATE3,STATE4);
BEGIN
PROCESS (CLOCK,CLEAR)
VARIABLE STATE : STATES;
BEGIN
IF ( CLEAR='0' ) THEN
STATE := IDLE;
DAC_CS <= '1';
TRANS_EN <= '0';
WGEN_SEL <= '0';
WGEN_ST <= '0';
ELSIF CLOCK='1' AND CLOCK'event THEN
CASE STATE IS
WHEN IDLE =>
IF (EN='1' AND (L_PULSE='1' OR H_PULSE='1')) OR (WGEN_BUSY='1') THEN
STATE := STATE0;
DAC_CS <= '0';
TRANS_EN <= '0';
WGEN_SEL <= WGEN_SEL;
WGEN_ST <= WGEN_ST;
ELSE
STATE := IDLE;
DAC_CS <= '1';
TRANS_EN <= '0';
WGEN_SEL <= WGEN_SEL;
WGEN_ST <= WGEN_ST;
END IF;
WHEN STATE0 =>
IF L_PULSE='1' AND WGEN_BUSY='0' THEN
STATE := STATE1A;

```

```

DAC_CS <= '0';
TRANS_EN <= '1';
WGEN_SEL <= '0';
WGEN_ST <= WGEN_ST;
ELSIF L_PULSE='0' AND H_PULSE='1' AND WGEN_BUSY='0' THEN
  STATE := STATE1B;
  DAC_CS <= '0';
  TRANS_EN <= '1';
  WGEN_SEL <= '1';
  WGEN_ST <= WGEN_ST;
ELSIF WGEN_BUSY='1' THEN
  STATE := STATE3;
  DAC_CS <= '0';
  TRANS_EN <= '1';
  WGEN_SEL <= WGEN_SEL;
  WGEN_ST <= WGEN_ST;
END IF;
WHEN STATE1A =>
  IF TRANS_END='1' THEN
    STATE := STATE2A;
    DAC_CS <= '1';
    TRANS_EN <= '0';
    WGEN_SEL <= '0';
    WGEN_ST <= '1';
  ELSE
    STATE := STATE1A;
    DAC_CS <= '0';
    TRANS_EN <= '1';
    WGEN_SEL <= '0';
    WGEN_ST <= WGEN_ST;
  END IF;
WHEN STATE2A =>
  STATE := IDLE;
  DAC_CS <= '1';
  TRANS_EN <= '0';
  WGEN_SEL <= WGEN_SEL;
  WGEN_ST <= WGEN_ST;
WHEN STATE1B =>
  IF TRANS_END='1' THEN
    STATE := STATE2B;
    DAC_CS <= '1';
    TRANS_EN <= '0';
    WGEN_SEL <= '1';
    WGEN_ST <= '1';
  ELSE
    STATE := STATE1B;
    DAC_CS <= '0';
    TRANS_EN <= '1';
    WGEN_SEL <= '1';
    WGEN_ST <= WGEN_ST;
  END IF;
WHEN STATE2B =>
  STATE := IDLE;
  DAC_CS <= '1';

```

```

TRANS_EN <= '0';
WGEN_SEL <= WGEN_SEL;
WGEN_ST <= WGEN_ST;
WHEN STATE3 =>
  IF TRANS_END='1' AND WGEN_BUSY='0' THEN
    STATE := STATE4;
    DAC_CS <= '1';
    TRANS_EN <= '0';
    WGEN_SEL <= WGEN_SEL;
    WGEN_ST <= '0';
  ELSE
    STATE := STATE3;
    DAC_CS <= '0';
    TRANS_EN <= '1';
    WGEN_SEL <= WGEN_SEL;
    WGEN_ST <= WGEN_ST;
  END IF;
WHEN STATE4 =>
  STATE := IDLE;
  DAC_CS <= '1';
  TRANS_EN <= '0';
  WGEN_SEL <= WGEN_SEL;
  WGEN_ST <= WGEN_ST;
END CASE;
END IF;
END PROCESS;
END BEHAVIOR;

```



```

RA      EQU      PORTA
RB      EQU      PORTB

DIN      EQU      D'0' ; (RA)
SCLK     EQU      D'1' ; (RA)
DOWN     EQU      D'2' ; (RA)
UP       EQU      D'3' ; (RA)
SYS_OFF  EQU      D'4' ; (RA)

SYS_ON   EQU      D'0' ; (RB)
DSR      EQU      D'1' ; (RB)
READY    EQU      D'2' ; (RB)
CS       EQU      D'3' ; (RB)
ANT_EN   EQU      D'4' ; (RB)
BAT_EN   EQU      D'5' ; (RB)
RB6      EQU      D'6' ; (RB)
RB7      EQU      D'7' ; (RB)

ComErr   EQU      D'0' ; (Flags)
CRCErr   EQU      D'1' ; (Flags)
Stdby    EQU      D'2' ; (Flags)
Shtdwn   EQU      D'3' ; (Flags)
NewData  EQU      D'4' ; (Flags)
SysOn    EQU      D'5' ; (Flags)

NbWordRead EQU      D'6'
NbParam   EQU      D'4'
ROMlength EQU      D'7'

_CtrlDAC0 EQU      0x00 ; offset en EEPROM(0) du param "CtrlDAC"
_Amplitude0 EQU      0x01 ; offset en EEPROM(0) du param "Amplitude"
_CtrlDAC1 EQU      0x07 ; offset en EEPROM(1) du param "CtrlDAC"
_Amplitude1 EQU      0x08 ; offset en EEPROM(1) du param "Amplitude"

;*****
;          DEFINITION des VARIABLES          *
;*****
Temp1     EQU      0x0C ; Vars locales pour sousroutines niveau 0
Temp2     EQU      0x0D ; Vars locales pour sousroutines niveau 0
Temp3     EQU      0x0E ; Vars locales pour sousroutines niveau 0
Temp4     EQU      0x0F ; Vars locales pour sousroutines niveau 0

NbStim    EQU      0x10
Nb128Stim EQU      0x11

Flags     EQU      0x12 ; b5=SysOn,b4=DataOK,b3=Shtdwn,b2=Stdby
; b1=CRCErr,b0=ComErr
ActiveROM EQU      0x13

RAM       EQU      0x20 ; Début de l'espace RAM
CmdWord   EQU      0x20
Checksum  EQU      0x25

CtrlDAC   EQU      0x20

```

```

Amplitude      EQU      0x21
PulseWide      EQU      0x22
Frequency       EQU      0x23
TimeOnOff      EQU      0x24
CtrlDACStdby   EQU      0x25
AmplitudeZero  EQU      0x26

```

```

;*****
;                                     DONNEES EEPROM STATIQUES
;*****
ORG      0x2100
DE       B'11101001' ; Amp ctrl byte           0x00
DE       D'39'       ; Amplitude (8 bits)       0x01
DE       D'24'       ; Pulse wide (8 bits)      0x02
DE       B'10000000' ; Frequency (2+2+4 bits)   0x03
DE       B'01001011' ; time to be ON and OFF (4+4 bits) 0x04
DE       B'11111001' ; Stdby ctrl byte         0x05
DE       D'0'        ; Amplitude zero          0x06

DE       B'11101001' ; Amp ctrl byte           0x07
DE       D'39'       ; Amplitude (8 bits)       0x08
DE       D'24'       ; Pulse wide (8 bits)      0x09
DE       B'10000000' ; Frequency (2+2+4 bits)   0x0A
DE       B'01001011' ; time to be ON and OFF (4+4 bits) 0x0B
DE       B'11111001' ; Stdby ctrl byte         0x0C
DE       D'0'        ; Amplitude zero          0x0D

;*****
;                                     INITIALISATION du VECTEUR de RESET
;*****
ORG      0x000
        goto      Main

;*****
;                                     CODE d'INTERRUPTION MATERIELLE
;*****
ORG      0x004
        return

;*****
;***** PROGRAMME PRINCIPAL *****
;*****
Main
        movlw     B'00010000',RA      ; Init Port A
        movlw     B'00010000',RB      ; Init Port B
        bank1
        clrf      RA                  ; Config Port A
        movlw     B'00000011',RB      ; Config Port B
        lo        OPTION_REG,T0CS
        bank0
        clrf      Flags
        clrf      ActiveROM

LoopMain
        clrwdt

```

```

SysOn=0      call    NoStim      ; Call "NoStim" if SYS_ON=0 and DSR=0 and
              call    HighZ      ; Call "HighZ" if SYS_ON=1 and DSR=0
              call    Communicate; Call "Communicate" if SYS_ON=1 and
DSR=1
and SysOn=1  call    Stimulate   ; Call "Stimulate" if SYS_ON=0 and DSR=0
              btfsc   Flags,Shtdwn
              goto    ShutDown
              goto    LoopMain
ShutDown     lo      RA,SYS_OFF
              movlw   B'00000110'
              call    Standby
              goto    LoopMain

;*****
;***** FIN du PROGRAMME PRINCIPAL *****
;*****

;*****
;***** NOSTIM SUBROUTINE [NIVEAU 1] *****
;
; Action      : Force le CPU en sommeil tant que SYS_ON=0 and DSR=0
; Entrées     : none
; Sousroutine : Standby
;*****
NoStim
              btfsc   RB,SYS_ON
              goto    EndNoStim
              btfsc   RB,DSR
              goto    EndNoStim
              btfsc   Flags,SysOn
              goto    EndNoStim
              movlw   B'01000111'
              hi      INTCON,INTE
              call    Standby
              btfss   INTCON,INTF
              goto    NoStim
EndNoStim
              clrw
              call    Standby
              return ; of NoStim subroutine

;*****
;***** HIGHZ SUBROUTINE [NIVEAU 1] *****
;
; Action      : Force le CPU en sommeil tant que SYS_ON=1 and DSR=0
;              Assure aussi la connexion du DAC à RF_PWR
; Entrées     : N/A
; Sousroutine : Standby
;*****
HighZ
              btfss   RB,SYS_ON
              goto    EndEndHighZ
              btfsc   RB,DSR

```

```

goto      EndEndHighZ
clrwf
call      Standby
btfss    RB,SYS_ON
goto      EndEndHighZ
btfsc    RB,DSR
goto      EndEndHighZ
bank1
movlw    B'00001111',RA      ; Config Port A
movlw    B'00001111',RB      ; Config Port B
bank0
clrf     RB                  ; Init Port B

LoopHighZ
btfss    RB,SYS_ON
goto      EndHighZ
btfsc    RB,DSR
goto      EndHighZ
movlw    B'00000111'
hi       INTCON,INTE
call     Standby
btfss    INTCON,INTF
goto     LoopHighZ

EndHighZ
movlw    B'00010000',RA      ; Init Port A
movlw    B'00010000',RB      ; Init Port B
bank1
clrf     RA                  ; Config Port A
movlw    B'00000011',RB      ; Config Port B
bank0

EndEndHighZ
return ; of HighZ subroutine

;*****
;               COMMUNICATE SUBROUTINE [NIVEAU 1]                *
; Action      : Établi la connexion avec la FPGA pour recevoir  *
;              les nouveaux paramètres de stimulation permanente *
; Entrées    : N/A                                              *
; Sousroutine : GetData,AnalyseData, RAM2ROM, Standby          *
;*****
Communicate
btfss    RB,SYS_ON
goto      EndCommunicate
btfss    RB,DSR
goto      EndCommunicate
clrwf
call     Standby
btfss    RB,SYS_ON
goto      EndCommunicate
btfss    RB,DSR
goto      EndCommunicate
bank1
movlw    B'00001101',RA      ; Config Port A
movlw    B'00001111',RB      ; Config Port B
bank0

```

```

movlw    RAM
call     GetData
movlwf   B'00010000',RA    ; Init Port A
movlwf   B'00010000',RB    ; Init Port B
bank1
movlwf   B'00000011',RB    ; Config Port B
clrf     RA                ; Config Port A
bank0
btfsc    Flags,ComErr
goto     WaitEndCommunicate
btfsc    Flags,CRCErr
goto     WaitEndCommunicate
movfw    CmdWord
call     AnalyseData
btfss    Flags,NewData
goto     WaitEndCommunicate
movlw    Amplitude
call     RAM2ROM
lo        Flags,NewData
WaitEndCommunicate
btfss    RB,SYS_ON
goto     EndCommunicate
btfss    RB,DSR
goto     EndCommunicate
movlw    B'00000111'
hi        INTCON,INTE
call     Standby
btfss    INTCON,INTF
goto     WaitEndCommunicate
EndCommunicate
return ; of Communicate subroutine

```

```

;*****
;                               STIMULATE SUBROUTINE [NIVEAU 1]          *
; Action   : Génère la stimulation permanente tant que SYS_ON=0        *
;           et DSR=0                                                    *
; Entrées  : N/A                                                         *
; Sousroutine : ROM2RAM, SendData, UpDown, InterPhases, TimeOff, Standby *
;*****
Stimulate

```

```

btfsc    RB,SYS_ON
goto     EndStimulate
btfsc    RB,DSR
goto     EndStimulate
btfss    Flags,SysOn
goto     EndStimulate
clrw
call     Standby
btfsc    RB,SYS_ON
goto     EndStimulate
btfsc    RB,DSR
goto     EndStimulate
btfss    Flags,SysOn
goto     EndStimulate

```

```

        movlw    CtrlDAC
        call     ROM2RAM
        lo       Amplitude,7           ; Sécurité pour forcer I
                                         ; à être inférieur à 1mA

        movlwf   D'128',NbStim
        swapf    TimeOnOff,W
        andlw    B'00001111'
        movwf    Nb128Stim
LoopStimulate
        clrwdt
        movlwf   B'00111000',RB       ; force
BAT_EN=1,ANT_EN=1,CS=1,READY=0
        movlw    CtrlDAC
        call     SendData
        movfw    PulseWide
        call     UpDown
        movlw    CtrlDACStdby
        call     SendData
        movfw    Frequency
        call     InterPhases
        decf     NbStim,F
        bnz      ContinueStimulate
        movlwf   D'128',NbStim
        decf     Nb128Stim,F
        bnz      ContinueStimulate
        movlwf   B'00010000',RB
        movfw    TimeOnOff
        call     TimeOff
        swapf    TimeOnOff,W
        andlw    B'00001111'
        movwf    Nb128Stim
ContinueStimulate
        btfss    RB,SYS_ON
        goto     LoopStimulate
        movlwf   B'00010000',RB
EndStimulate
        return  ; of Stimulate subroutine

;=====
;=====
;*****
;               STANDBY SUBROUTINE [BAS NIVEAU]                *
; Action       : Force le CPU en standby pendant X secondes   *
; Entrées      : W[6]=Interrupt edge, W[2..0]=WDT prescaler bits *
; Registres:    OPTION_REG,INTCON                               *
;*****
Standby
        bank1
        iorlw    B'10111000'
        movwf    OPTION_REG
        lo       INTCON,INTF
        clrwdt
        nop

```

```

        sleep                ; force CPU en mode standby
        nop
        lo      INTCON, INTE
        movlwf  B'11011111', OPTION_REG
        bank0
EndStandby
        return ; of Standby subroutine

;=====
;=====

;*****
;                               GETDATA SUBROUTINE [NIVEAU 0]                *
; Action      : Réceptionne les données du FPGA via DIN                    *
;              et les transfert en RAM                                     *
; Entrées     : W = pointer en RAM                                         *
; Registres:  Temp1, Temp2, Temp3, Temp4, Flags, FSR, INDF                 *
; Rq          : Doit nécessairement être en mode "programmation"          *
;*****
GetData
        movwf   FSR                ; Initialise le pointeur en RAM
        clrf    Flags
        movlwf  NbWordRead, Temp3
        movlwf  D'008', Temp2
        clrf    Temp4
LoopGetData
        btfss   RB, SYS_ON
        goto    ErrorComGetData
        btfss   RB, DSR             ; Test si synchro OK
        goto    ErrorComGetData
        rlf     Temp1, F
        btfss   RA, DIN              ; Temp1[0]=0 if DIN=0
        lo      Temp1, 0
        btfsc   RA, DIN              ; Temp1[0]=1 if DIN=1
        hi      Temp1, 0
        hi      RA, SCLK             ; Clock Registres du FPGA
        lo      RA, SCLK
        decfsz  Temp2, F
        goto    LoopGetData
        movf    Temp1, F
        bz      ErrorComGetData
        movfw   Temp1
        movwf   INDF
        decf    Temp3, F
        bz      CheckSumGetData
        addwf   Temp4, F
        incf    FSR, F
        movlwf  D'008', Temp2
        goto    LoopGetData
CheckSumGetData
        subwf   Temp4, W
        skpz
        hi      Flags, CRCErr
EndGetData

```

```

        return ; of GetData subroutine
ErrorComGetData
        hi      Flags,ComErr
        goto    EndGetData

;*****
;
;           ANALYSEDATA SUBROUTINE [NIVEAU 0]
; Action   : Analyse les données réceptionnées du FPGA
; Entrées  : W = mot de commande reçu du FPGA
; Registres: Temp1,Flags
;*****
AnalyseData
        andlw   B'11111111'
        bz      EndAnalyseData
        movwf   Temp1
        incf    Temp1,W
        bz      EndAnalyseData
        swapf   Temp1,W
        xorwf   Temp1,W          ; Compare les nibbles
        bnz     EndAnalyseData ; Termine si nibbles non identique
        rlf     Temp1,F
        rlf     Temp1,W
        xorlw   B'00111100'
        andlw   B'00111100'
        movwf   Flags
EndAnalyseData
        return ; of AnalyseData subroutine

;*****
;
;           RAM2ROM SUBROUTINE [NIVEAU 0]
; Action   : Transfère les données de la RAM vers la EEPROM
; Entrées  : W = pointeur en RAM
; Registres: Temp1,FSR,INDF,EEADR,EEDATA,EECON1,EECON2
;*****
RAM2ROM
        movwf   FSR
        movlwf  NbParam,Temp1
        movlwf  _Amplitude0,EEADR
        btfsc   ActiveROM,0
        movlwf  _Amplitude1,EEADR ; initialise le pointer EEPROM
Loop1RAM2ROM
        clrw    INDF
        movfw   EEDATA
        bankl   EECON1,WREN
        movlwf  H'55',EECON2      ; active écriture EEPROM
        movlwf  H'AA',EECON2      ; active écriture EEPROM
        hi      EECON1,WR
Loop2RAM2ROM
        btfss   EECON1,EEIF      ; test si écriture OK
        goto    Loop2RAM2ROM
        lo      EECON1,EEIF
        lo      EECON1,WREN

```

```

bank0
decf    Temp1,F
bz      EndRAM2ROM
incf    FSR,F          ; pointe la donnée suivant en RAM
incf    EEADR,F        ; pointe la donnée suivant en
EEPROM
goto    Loop1RAM2ROM
EndRAM2ROM
return ; of RAM2ROM subroutine

;=====
;=====
;*****
;
;          ROM2RAM SUBROUTINE [NIVEAU 0]
; Action   : Transfère les données de la EEPROM vers la RAM
; Entrées  : W = pointer en RAM
; Registres: Temp1,FSR,INDF,EEADR,EEDATA,EECON1
;*****
ROM2RAM
movwf   FSR          ; initialise le pointer RAM
movlwf  _CtrlDAC0,EEADR ; initialise le pointer EEPROM
movlwf  ROMlength,Temp1
LoopROM2RAM
bank1
hi      EECON1,RD
bank0
movfw   EEDATA
movwf   INDF
decf    Temp1,F
bz      EndROM2RAM
incf    FSR,F          ; pointe la donnée suivant en RAM
incf    EEADR,F        ; pointe la donnée suivant en
EEPROM
goto    LoopROM2RAM
EndROM2RAM
return ; of ROM2RAM subroutine

;*****
;
;          SENDDATA SUBROUTINE [NIVEAU 0]
; Action   : Transmet les mots de commande pointés par W au DAC
;           avec contrôle sur SCLK and CS
; Entrées  : W = pointer en RAM
; Registres: Temp1,Temp2,Temp3,FSR,INDF
; Rq       : Doit nécessairement être en mode "Stimulation permanente"
;*****
SendData
movwf   FSR          ; initialise le pointer RAM
lo      RB,CS        ; force CS=0
movlwf  D'002',Temp2 ; 2 mots par commande
Loop2SendData
movlwf  D'008',Temp1 ; 8 bits par mot
movfw   INDF
movwf   Temp3

```

```

Loop1SendData
    btfss    Temp3,7
    lo       RA,DIN          ; DIN=0 if INDF[7]=0
    btfsc    Temp3,7
    hi       RA,DIN          ; DIN=1 if INDF[7]=1
    hi       RA,SCLK         ; force SCLK=1
    lo       RA,SCLK         ; force SCLK=0
    rlf      Temp3,F
    decfsz   Temp1,F
    goto     Loop1SendData
    incf     FSR,F           ; incrémente le pointeur en RAM
    decfsz   Temp2,F
    goto     Loop2SendData
    hi       RB,CS           ; force CS=1
    lo       RA,DIN          ; force DIN=0

EndSendData
    return ; of SendData subroutine

;*****
;                               UPDOWN SUBROUTINE [NIVEAU 0]
; Action   : Contrôle les signaux UP et DOWN pour la commutation
;           du courant
; Entrées  : W = durée d'une impulsion
; Registres: Temp1,Temp2
; Rq       : Doit nécessairement être en mode "Stimulation permanente"
;*****
UpDown
    movwf    Temp1
    movwf    Temp2
    movlw    B'00011000'
    hi       RA,DOWN         ; force UP=0 and DOWN=1

Loop1UpDown
    decfsz   Temp1,F
    goto     Loop1UpDown
    movwf    RA              ; force UP=1 and DOWN=0

Loop2UpDown
    decfsz   Temp2,F
    goto     Loop2UpDown
    lo       RA,UP           ; force UP=0 and DOWN=0

EndUpDown
    return ; of UpDown subroutine

;*****
;                               INTERPHASES SUBROUTINE [NIVEAU 0]
; Action   : Contrôle la période entre 2 impulsions
; Entrées  : W[7..6]=[PS1,PS0] of prescaler bits, W[5..4]=additional
;           single sleep delay, W[3..0]=timer0 repeat time
; Registres: Temp1,Temp2,Temp3,OPTION_REG,INTCON,TMR0
;*****
InterPhases
    movwf    Temp1
    movwf    Temp2
    andlw    B'11000000'
    bz       AdditionalSleep

```

MainSleep

```

    clrc
    rrf    Temp1, F
    rrf    Temp1, F
    swapf  Temp1, F
    decf   Temp1, W
    iorlw  B'11111000'
    btfsc  RB, SYS_ON
    goto   EndInterPhases
    hi     INTCON, INTE
    call   Standby
    btfsc  INTCON, INTF
    goto   EndInterPhases

```

AdditionalSleep

```

    swapf  Temp2, W
    andlw  B'00000011'
    bz     EpsilonWait
    movwf  Temp3

```

LoopAddSleep

```

    btfsc  RB, SYS_ON
    goto   EndInterPhases
    movlw  B'11111000'
    hi     INTCON, INTE
    call   Standby
    btfsc  INTCON, INTF
    goto   EndInterPhases
    decfsz Temp3, F
    goto   LoopAddSleep

```

EpsilonWait

```

    rlf    Temp2, F
    movlw  B'00011110'
    andwf  Temp2, F
    bz     EndInterPhases

```

Loop1EpsilonWait

```

    clrf   TMR0
    lo     INTCON, T0IF

```

Loop2EpsilonWait

```

    btfsc  RB, SYS_ON
    goto   EndInterPhases
    btfss  INTCON, T0IF
    goto   Loop2EpsilonWait
    decfsz Temp2, F
    goto   Loop1EpsilonWait

```

EndInterPhases

```

    return ; of InterPhases subroutine

```

```

;*****
;                               TIMEOFF SUBROUTINE [NIVEAU 0]
; Action   : Contrôle la durée "éteint" entre 2 train d'impulsions
; Entrées  : W[3..2]=repeated sleep, W[1..0]=[PS1,PS0] prescaler bits
; Registres: Temp1, OPTION_REG, INTCON
;*****
TimeOff

```

```

    bank1

```

```

        movwf    Temp1
        iorlw    B'11111100'
        movwf    OPTION_REG
        bank0
        rrf      Temp1,F
        movlw    B'00000110'
        andwf    Temp1,F
        hi       INTCON,INTE
LoopTimeOff
        btfsc    RB,SYS_ON
        goto     ContinueTimeOff
        lo       INTCON,INTF
        clrwdt
        nop
        sleep
        nop
        btfsc    INTCON,INTF
        goto     ContinueTimeOff
        movf     Temp1,F
        bz       ContinueTimeOff
        decfsz   Temp1,F
        goto     LoopTimeOff
ContinueTimeOff
        bank1
        lo       INTCON,INTE
        movlwf   B'11011111',OPTION_REG
        bank0
EndTimeOff
        return ; of TimeOff subroutine

END

```

ANNEXE C

SCHÉMAS ET CODES DES CONTRÔLEURS EXTERNES

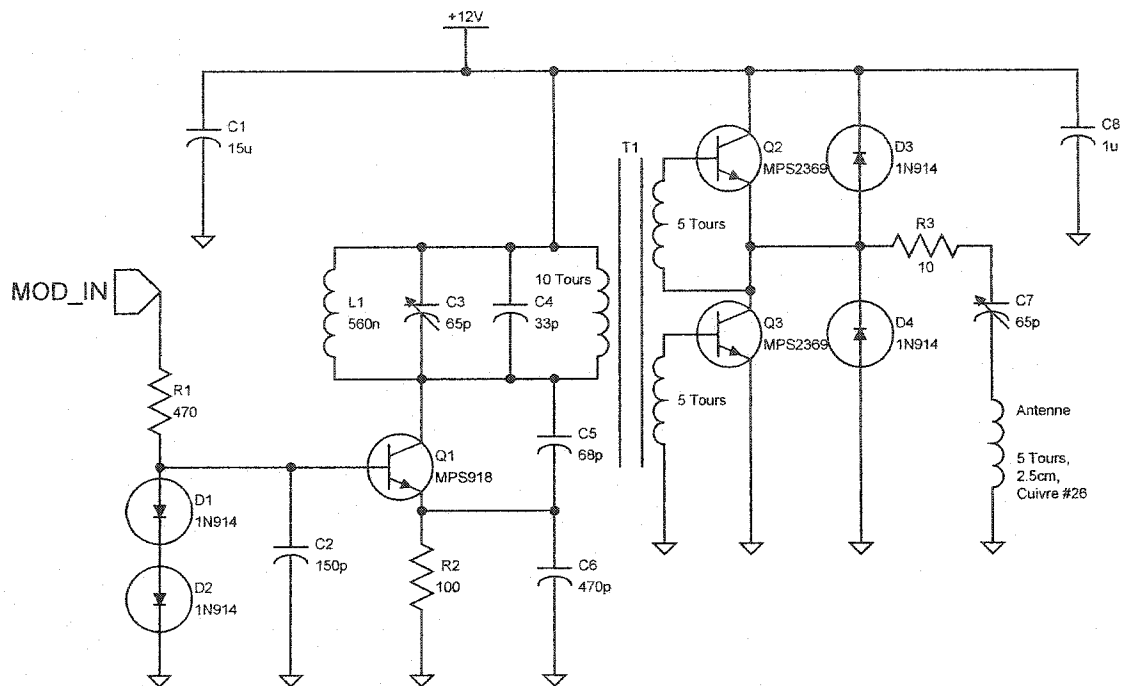


**Figure C.1 : Schéma électrique de l'émetteur RF
utilisé pour les contrôleurs externes**

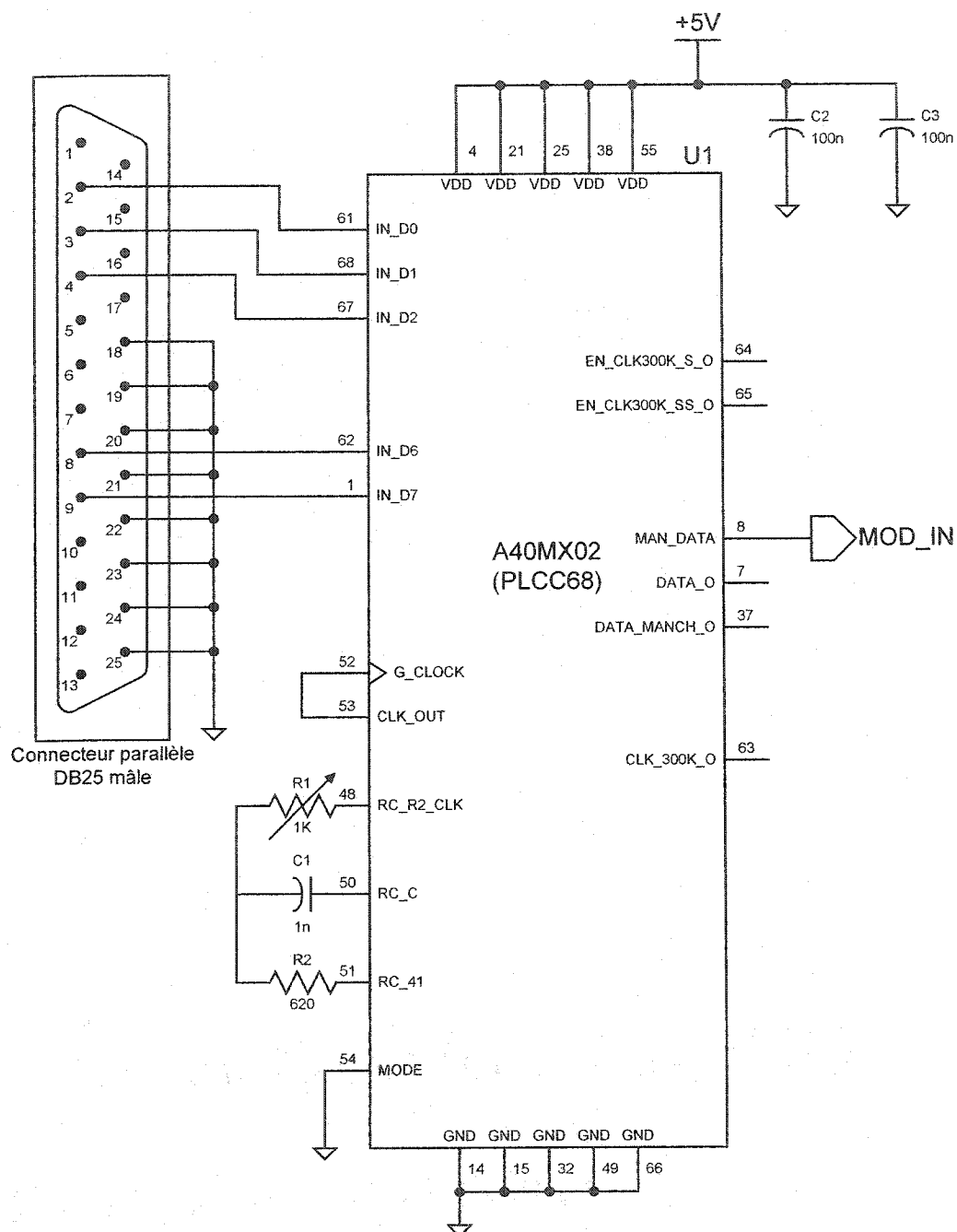


Figure C.2 : Schéma électrique de l'interface parallèle du CTF

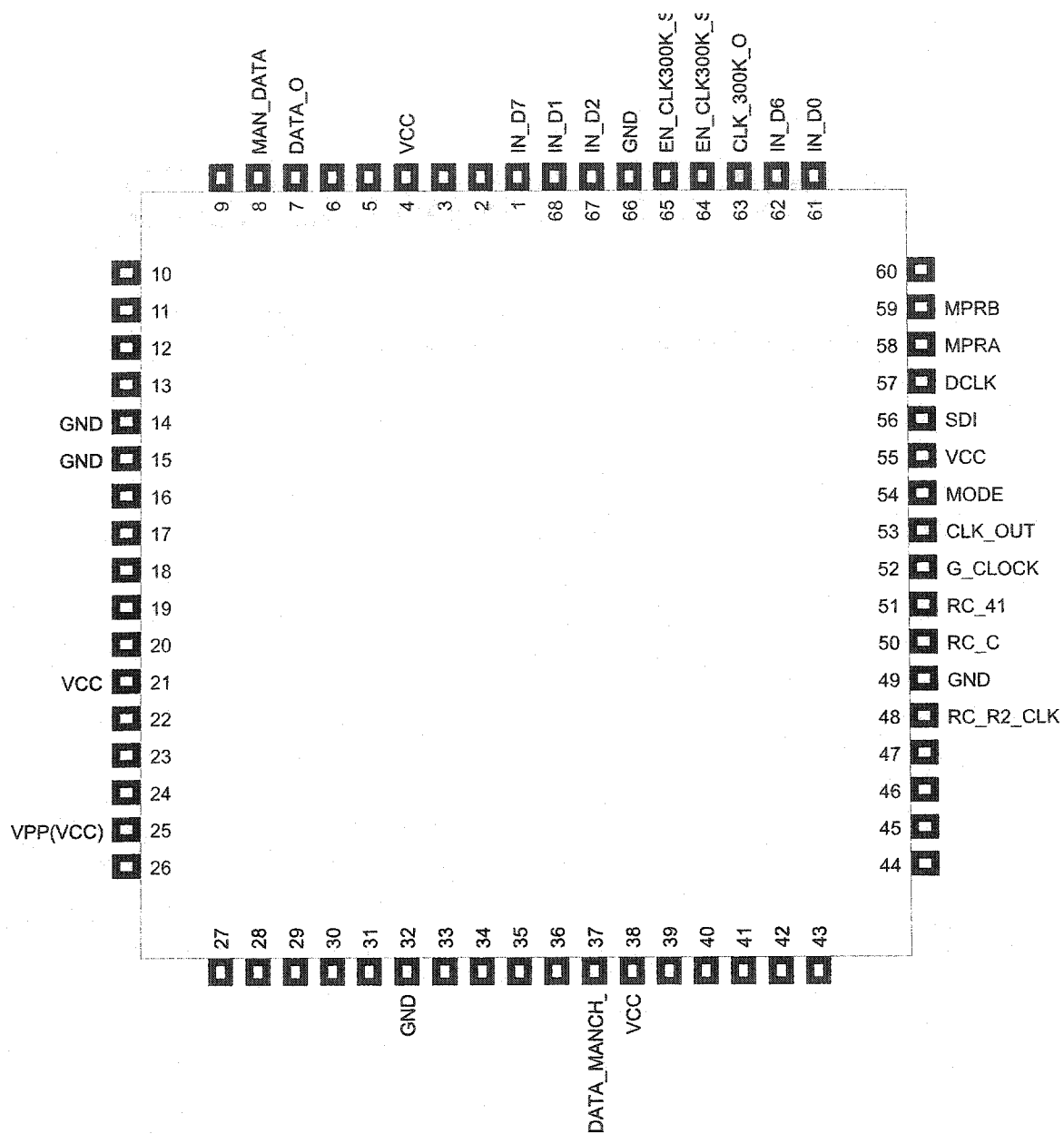


Figure C.3 : Répartition des broches du FPGA de l'interface parallèle du CTF

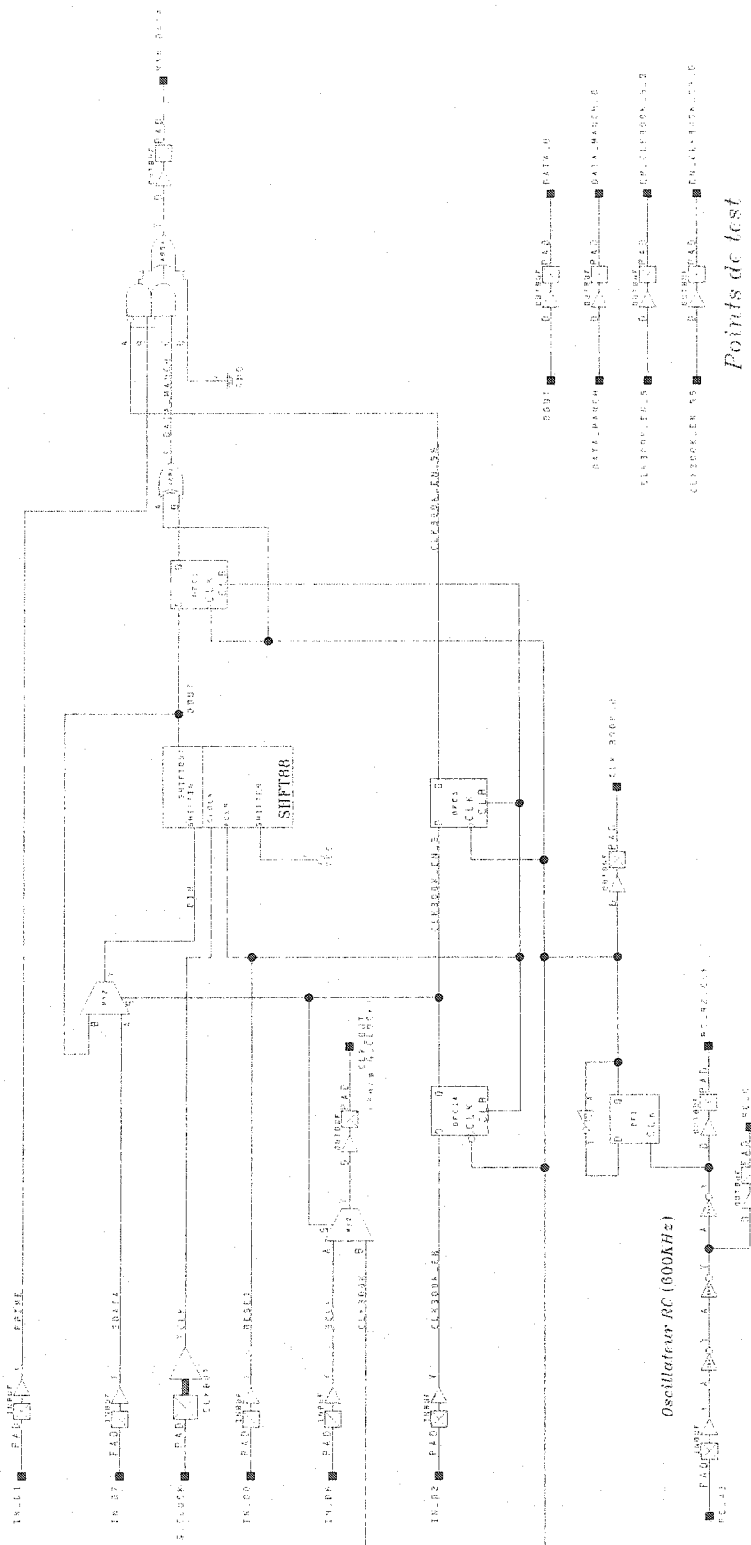


Figure C.4 : Schéma logique de niveau hiérarchique supérieur du PFGA de l'interface parallèle du CTF

```

/*****
*   Nom du fichier : StimR3.c
*   Date           : 20 Nov 2000
*   Version        : 3.4
*   Auteur         : E. Schneider
*****/
*   Fichiers requis : stdio.h, conio.h, string.h, dos.h, time.h, stdlib.h
*****/
*   Notes: Programme C du logiciel PC du CTF
*           D0 = Reset
*           D1 = Prime
*           D2 = Clk300K_EN
*           D6 = SCLK
*           D7 = Sdata
*****/

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <dos.h>
#include <time.h>
#include <stdlib.h>

typedef unsigned int WORD;

#define GAUCHE 0x14B
#define DROIT  0x14D
#define END    0x14F
#define HOME   0x147
#define ENTER  0x00D
#define ESC    0x01B
#define POINT  0x02E
#define F1     0x13B
#define F2     0x13C
#define F3     0x13D
#define F4     0x13E
#define ALTX   0x12D
#define ALTR   0x113

#define C1  3
#define C2  45
#define CC  4
#define L1  6
#define L2  8
#define L3  10
#define L4  12
#define L5  14
#define L6  16

#define X1  3
#define X2  30

```

```

#define X3 55
#define Y1 18
#define Y2 25
#define XCur 5
#define YCur 23

#define TEMPO 300

#define entete 0x7E
#define cmd_sel 0x99
#define cmd_perm 0xAA
#define cmd_imp 0xBB
#define cmd_reset 0x55
#define sdi_max 10

#ifdef MK_FP //Macro MK_FP déjà définie ?
    #undef MK_FP // Oui on l'efface
#endif
#define WK if(!(getch())) getch()
#define MK_FP(seg,ofs) ((void far *) ((unsigned long) (seg)<<16|(ofs)))

typedef struct tval {
    unsigned char nom[20];
    float val;
    unsigned int bin;
    unsigned char length;
    unsigned char val_bin;
    unsigned char posx,posy;
    unsigned char offset;
    unsigned char unite[5];
} tval;

int GetPortAdr( int Numero );
void wait(int t);
void auto_test(void);
void init(void);
void printbin(unsigned int bin,unsigned char length);
void redraw(void);
void disp_cmd_pic(tval *d);
void change_cmd_pic(tval *d);
void calc_on_off(tval *tof,tval *freq,float *on,float *off);
void disp_time_on_off(tval *tof,tval *freq);
void change_time_on_off(tval *tof,tval *freq);
void data_stim(void);
void data_imp(void);
void data_perm(void);
int serialize_data(void);
void view(void);
void send(void);
void reset(void);
void prime(void);
void sd_reset(void);
int rdffloat(float *val,float valini);
int txt2float(char *src,float *dest);

```



```

        break;
    case F2 : mode=6;
        break;
    case F3 : mode=12;
        break;
    case F4 : auto_test();
        break;
    case ALT_X : quitte=1;
        reset();
        outportb(donnees, 0x00);
        break;
    }
    if (mode==0) {
        data_stim();
    } else if (mode==6) {
        data_imp();
    } else if (mode==12) {
        data_perm();
    }
    redraw();
}
clrscr();
}

int GetPortAdr( int Numero ) {
// Lit les adresses des ports dans le segment de variables du BIOS
donnees = *( WORD far * ) MK_FP( 0x0040, 6 + Numero * 2 );
if ( donnees != 0 ) { // Interface disponible ?
    etat = donnees + 1; // Adresse pour le registre d'état
    commande = donnees + 2; // Adresse pour le registre de commandes
    return 1; // Pas d'erreur
} else {
    return 0; //Erreur: interface absente
}
}

void wait(int t) {

time_t t0;

    t0=time(NULL);
    while ((time(NULL)-t0)<t && !kbhit()) {
        gotoxy(1,1);
        printf("%d",time(NULL)-t0);
    }
    gotoxy(1,1);
    clreol();
}

void auto_test(void) {

time_t t0;
unsigned char num_test;

```

```

while (!kbhit()) {
    num_test=random(8);
    switch (num_test) {
    case 0 : // Stimulation sélective
        data[0].val=1.1;
        data[1].val=1.5;
        data[2].val=175.0;
        data[3].val=100.0;
        data[4].val=30.0;
        data[5].val=600.0;
        mode=0;
        data_stim();
        redraw();
        send();
        wait(60);
        reset();
        wait(60);
        break;
    case 1 : // stimulation sélective
        data[0].val=0.9;
        data[1].val=1.5;
        data[2].val=175.0;
        data[3].val=60.0;
        data[4].val=30.0;
        data[5].val=600.0;
        mode=0;
        data_stim();
        redraw();
        send();
        wait(60);
        reset();
        wait(5);
        break;
    case 2 : // stimulation sélective
        data[0].val=2.0;
        data[1].val=0.0;
        data[2].val=210.0;
        data[3].val=0.0;
        data[4].val=30.0;
        data[5].val=0.0;
        mode=0;
        data_stim();
        redraw();
        send();
        wait(60);
        reset();
        wait(30);
        break;
    case 3 : // stimulation sélective
        data[0].val=0.9;
        data[1].val=0.0;
        data[2].val=175.0;
        data[3].val=0.0;
        data[4].val=30.0;

```

```

data[5].val=0.0;
mode=0;
data_stim();
redraw();
send();
wait(60);
reset();
wait(5);
break;
case 4 : // Stimulation permanente
data[12].bin=0x33;
data[13].val=200.0;
data[14].val=150.0;
data[15].val=20.8;
data[16].bin=0x80;
mode=12;
data_perm();
redraw();
send();
wait(10);
reset();
wait(900);
break;
case 5 : // Stimulation permanente
data[12].bin=0x33;
data[13].val=400.0;
data[14].val=300.0;
data[15].val=8.3;
data[16].bin=0xF3;
mode=12;
data_perm();
redraw();
send();
wait(10);
reset();
wait(900);
break;
case 6 : // Stimulation permanente
data[12].bin=0x33;
data[13].val=900.0;
data[14].val=50.0;
data[15].val=39.0;
data[16].bin=0x80;
mode=12;
data_perm();
redraw();
send();
wait(10);
reset();
wait(900);
break;
case 7 : // Stimulation permanente
data[12].bin=0x33;
data[13].val=700.0;

```

```

        data[14].val=1000.0;
        data[15].val=50.0;
        data[16].bin=0xF3;
        mode=12;
        data_perm();
        redraw();
        send();
        wait(10);
        reset();
        wait(900);
        break;
    }
}

void init(void) {
    outportb(commande, 0xC4);
    reset(); //reset
    outportb(donnees, 0x00);

    mode=0;
    last_send="None";

    // paramètres de stimulation sélective
    strcpy(data[0].nom, "LFA");
    data[0].val=1.5;
    data[0].length=8;
    data[0].val_bin=0;
    data[0].posx=C1;
    data[0].posy=L2;
    data[0].offset=3+CC;
    strcpy(data[0].unite, "mA");

    strcpy(data[1].nom, "HFA");
    data[1].val=1.0;
    data[1].length=8;
    data[1].val_bin=0;
    data[1].posx=C2;
    data[1].posy=L2;
    data[1].offset=3+CC;
    strcpy(data[1].unite, "mA");

    strcpy(data[2].nom, "LFW");
    data[2].val=200.0;
    data[2].length=6;
    data[2].val_bin=0;
    data[2].posx=C1;
    data[2].posy=L3;
    data[2].offset=3+CC;
    strcpy(data[2].unite, "as");

    strcpy(data[3].nom, "HFW");
    data[3].val=100.0;

```

```

data[3].length=6;
data[3].val_bin=0;
data[3].posx=C2;
data[3].posy=L3;
data[3].offset=3+CC;
strcpy(data[3].unite,"s");

strcpy(data[4].nom,"LFP");
data[4].val=30.0;
data[4].length=14;
data[4].val_bin=0;
data[4].posx=C1;
data[4].posy=L4;
data[4].offset=3+CC;
strcpy(data[4].unite,"Hz");

strcpy(data[5].nom,"HFP");
data[5].val=600.0;
data[5].length=10;
data[5].val_bin=0;
data[5].posx=C2;
data[5].posy=L4;
data[5].offset=3+CC;
strcpy(data[5].unite,"Hz");

// paramètres de mesure d'impédance
strcpy(data[6].nom,"Blank");
data[6].bin=0x00;
data[6].length=8;
data[6].val_bin=1;
data[6].posx=C1;
data[6].posy=L2;
data[6].offset=5+CC;
strcpy(data[6].unite,"");

strcpy(data[7].nom,"Ampl");
data[7].val=500.0;
data[7].length=8;
data[7].val_bin=0;
data[7].posx=C2;
data[7].posy=L2;
data[7].offset=4+CC;
strcpy(data[7].unite,"A");

strcpy(data[8].nom,"Blank");
data[8].bin=0x00;
data[8].length=6;
data[8].val_bin=1;
data[8].posx=C1;
data[8].posy=L3;
data[8].offset=5+CC;
strcpy(data[8].unite,"");

strcpy(data[9].nom,"Freq");

```

```

data[9].val=10.0;
data[9].length=6;
data[9].val_bin=0;
data[9].posx=C2;
data[9].posy=L3;
data[9].offset=4+CC;
strcpy(data[9].unite, "KHz");

strcpy(data[10].nom, "Timer1");
data[10].val=500.0;
data[10].length=14;
data[10].val_bin=0;
data[10].posx=C1;
data[10].posy=L4;
data[10].offset=6+CC;
strcpy(data[10].unite, "µs");

strcpy(data[11].nom, "Timer2");
data[11].val=500.0;
data[11].length=10;
data[11].val_bin=0;
data[11].posx=C2;
data[11].posy=L4;
data[11].offset=6+CC;
strcpy(data[11].unite, "µs");

// Paramètres de stimulation permanente
strcpy(data[12].nom, "Cmd PIC");
data[12].bin=0x33;
data[12].length=8;
data[12].val_bin=1;
data[12].posx=C1;
data[12].posy=L2;
data[12].offset=7+CC;
strcpy(data[12].unite, "");

strcpy(data[13].nom, "Amplitude");
data[13].val=500.0;
data[13].length=8;
data[13].val_bin=0;
data[13].posx=C2;
data[13].posy=L2;
data[13].offset=9+CC;
strcpy(data[13].unite, "µA");

strcpy(data[14].nom, "PulseWide");
data[14].val=300.0;
data[14].length=8;
data[14].val_bin=0;
data[14].posx=C1;
data[14].posy=L3;
data[14].offset=9+CC;
strcpy(data[14].unite, "µs");

```

```

strcpy(data[15].nom, "Frequency");
data[15].val=20.8;
data[15].length=8;
data[15].val_bin=0;
data[15].posx=C2;
data[15].posy=L3;
data[15].offset=9+CC;
strcpy(data[15].unite, "Hz");

strcpy(data[16].nom, "TimeOn/Off");
data[16].bin=0x70;
data[16].length=8;
data[16].val_bin=1;
data[16].posx=C1;
data[16].posy=L4;
data[16].offset=10+CC;
strcpy(data[16].unite, "");

strcpy(data[17].nom, "Checksum");
data[17].bin=0x00;
data[17].length=12;
data[17].val_bin=1;
data[17].posx=C2;
data[17].posy=L4;
data[17].offset=8+CC;
strcpy(data[17].unite, "");
}

void printbin(unsigned int bin,unsigned char length) {

    if ((wherex()+length)>80) { gotoxy(C1-1,wherey()+1); }
    gotoxy(wherex()+length,wherey());
    while (length!=0) {
        printf("%X",bin & 0x1);
        bin=bin>>1;
        length--;
        gotoxy(wherex()-2,wherey());
    }
}

void disp_cmd_pic(tval *d) {

    gotoxy((*d).posx+(*d).offset, (*d).posy);
    switch ((int) (*d).bin) {
        case 0x77 : printf("ON ");
                    break;
        case 0xBB : printf("NEW+OFF ");
                    break;
        case 0x33 : printf("NEW+ON ");
                    break;
        case 0xDD : printf("OFF ");
                    break;
        case 0xEE : printf("STANDBY ");
                    break;
    }
}

```

```

    }
}

void change_cmd_pic(tval *d) {

    switch ((int)(*d).bin) {
        case 0xEE : (*d).bin=0x77;
                    break;
        case 0x77 : (*d).bin=0xBB;
                    break;
        case 0xBB : (*d).bin=0x33;
                    break;
        case 0x33 : (*d).bin=0xDD;
                    break;
        case 0xDD : (*d).bin=0xEE;
                    break;
    }
}

void calc_on_off(tval *tof, tval *freq, float *ton, float *toff) {

    if ((*freq).val==0) { *ton=0.0; }
    else { *ton=(1/(*freq).val)*128.*(float)((*tof).bin>>4); }
    *toff=(float)(0x10<<(((*tof).bin & 0x03))*24e-3;
    if (((*tof).bin & 0x0C)!=0) { *toff=*toff * (float)(((*tof).bin &
0x0C)>>1); }
}

void disp_time_on_off(tval *tof, tval *freq) {

    float on,off;

    gotoxy((*tof).posx+(*tof).offset,(*tof).posy);
    calc_on_off(tof,freq,&on,&off);
    printf("%1.1f / %1.1f ",on,off);
    printbin((*tof).bin,(*tof).length);
    printf("(");
    gotoxy(wherex()+(*tof).length,wherey());
    printf(")");
}

void change_time_on_off(tval *tof, tval *freq) {

    float on,off;
    int tempR1,tempPS1;
    unsigned char tempR2,tempPS2;

    calc_on_off(tof,freq,&on,&off);
    gotoxy((*tof).posx+(*tof).offset,(*tof).posy);
    rdfloat(&on,on);
    gotoxy((*tof).posx+(*tof).offset,(*tof).posy);
    printf("%1.1f / ",on);
    rdfloat(&off,off);
    off=off/24e-3;
}

```

```

tempR1=1;
tempR2=0;
tempPS1=16;
tempPS2=0;
while ((int)off>(tempR1*tempPS1)) {
    tempR1=tempR1+2;
    tempR2++;
    if (tempR1==3) { tempR1=2; }
    if (tempR1==8) { tempR1=1; tempPS1<=1; tempR2=0; tempPS2++; }
}
(*tof).bin=((int)((on*(*freq).val)/128.)<<4)+(tempR2<<2)+tempPS2;
}

void redraw(void) {

int c;

clrscr();

gotoxy(28,2);
printf("      POLYSTIM");
gotoxy(28,3);
printf("PC controller interface");

for (c=0;c<=5;c++) {
    gotoxy(data[c+mode].posx,data[c+mode].posy);
    printf("%d-%s: ",c+1,data[c+mode].nom);
    if (data[c+mode].val_bin==0) {
        gotoxy(data[c+mode].posx+data[c+mode].offset,data[c+mode].posy);
        printf("%1.1f",data[c+mode].val);
        printf(" %s ",data[c+mode].unite);
    }
    printbin(data[c+mode].bin,data[c+mode].length);
    printf("(");
    gotoxy(wherex()+data[c+mode].length,wherey());
    printf(")");
}

gotoxy(C1,L1);
printf("Mode: ");
if (mode==0) {
    printf("Selective Stimulation");
    gotoxy(C2,L1);
    printf("Cmd FPGA: 1001 1001");
} else if (mode==6) {
    printf("Impedance Mesure");
    gotoxy(C2,L1);
    printf("Cmd FPGA: 1011 1011");
} else if (mode==12){
    printf("Permanente Stimulation");
    gotoxy(C2,L1);
    printf("Cmd FPGA: 1010 1010");
    disp_cmd_pic(&data[0+mode]);
    printbin(data[0+mode].bin,data[0+mode].length);
}

```

```

    printf("(");
    gotoxy(wherex()+data[0+mode].length,wherey());
    printf(")");
    disp_time_on_off(&data[4+mode],&data[3+mode]);
} else {}

gotoxy(X1,Y1);
printf("F1: Selective Stim");
gotoxy(X2,Y1);
printf("F2: Impedance Mesure");
gotoxy(X3,Y1);
printf("F3: Permanente Stim");

gotoxy(X1,Y1+1);
printf("ALT+R: Send Reset cmd");
gotoxy(X2,Y1+1);
printf("##: Edit Value");
gotoxy(X3,Y1+1);
printf("R : Reset controller");

gotoxy(X1,Y1+2);
if ((inportb(donnees)&(0x02))==0) {
    printf("P : Prime ON");
} else {
    printf("P : Prime OFF");
}
gotoxy(X2,Y1+2);
printf("ALT+X: Quit");

gotoxy(X1,Y1+3);
printf("SPACE bar: Send data to controller");

gotoxy(X1,Y2);
clreol();
printf("PRN#: %X      ",donnees);

printf("Last send: %s      ",last_send);

if ((inportb(donnees)&(0x02))==0) {
    printf("Prime: OFF      ");
} else {
    printf("Prime: ON      ");
}

if ((inportb(donnees)&(0x04))==0) {
    printf("Transmitter: Stop'd");
} else {
    printf("Transmitter: Active");
}

gotoxy(XCur,YCur);
clreol();
printf("Command: ");
}

```

```

void data_stim(void) {

unsigned int lfa,hfa,lfw,hfw,lfh,hfp;

    lfa=(int)((255)*(data[0].val/2.));
    hfa=(int)((255)*(data[1].val/2.));
    lfw=(int)(data[2].val*0.3);
    hfw=(int)(data[3].val*0.3);
    if (data[4].val==0.0) { lfp=0;}
    else { lfp=(int)(300000./data[4].val);}
    if (data[5].val==0.0) { hfp=0;}
    else { hfp=(int)(300000./data[5].val);}

    data[0].bin=lfa;
    data[1].bin=hfa;
    data[2].bin=lfw;
    data[3].bin=hfw;
    data[4].bin=lfh;
    data[5].bin=hfp;

    sdata[0]=entete;
    sdata[1]=cmd_sel;

}

void data_imp(void) {

unsigned int ampl,freq,timer1,timer2;

    ampl=(int)((255)*(data[7].val/2000.));
    if (data[9].val==0.0) { freq=0;}
    else { freq=(int)(300./data[9].val);}
    timer1=(int)(data[10].val*300.);
    timer2=(int)(data[11].val*300.);

    data[7].bin=ampl;
    data[9].bin=freq;
    data[10].bin=timer1;
    data[11].bin=timer2;

    sdata[0]=entete;
    sdata[1]=cmd_imp;

}

void data_perm(void) {

unsigned int cmd,ampl,pw,freq,tof,quotien,reste;
unsigned char checksum;

    ampl=(int)((255)*(data[13].val/2000.));

```

```

pw=(int)(data[14].val/8.4);
if (data[15].val==0) {freq=0;}
else {
    quotient=(int)((1/data[15].val)*1e3)/24;
    if (quotien>=4) {freq=3; freq<=2; freq=freq+quotien-4; }
    if (quotien==3) {freq=0x09;}
    if (quotien<3) {freq=quotien; freq<=2;}
    freq<=4;
    reste=(int)((long)((1/data[15].val)*1e6)%24000);
    reste=reste/(int)(535.*2.8);
    if (reste<16) { freq=freq+reste; }
    else { freq=freq+0x0F; }
}

checksum=(char)data[12].bin+(char)ampl+(char)pw+(char)freq+(char)data[1
6].bin;

data[13].bin=ampl;
data[14].bin=pw;
data[15].bin=freq;
data[17].bin=((int)checksum)<<4;

sdata[0]=entete;
sdata[1]=cmd_perm;

}

int serialize_data(void) {

unsigned int dbi;
unsigned char i,sdi,di,sdbi;
unsigned char cingl,data_end,sdata_end;
unsigned char crc7;

    crc=0; //Polynome=X^8+1
    sdi=1;
    sdbi=0x80;
    for (i=0;i<8;i++) {
        if ((sdata[sdi] & sdbi)==0) { // sdata[sdi][sdbi]=0
            cingl=0;
            crc7=(crc & 0x80);
            crc<=1;
            crc=crc & 0xFE;
            crc=crc | ((crc7>>7)^0x00);
            sdbi>>=1;
        } else { // sdata[sdi][sdbi]=1
            cingl++;
            crc7=(crc & 0x80);
            crc<=1;
            crc=crc & 0xFE;
            crc=crc | ((crc7>>7)^0x01);
            sdbi>>=1;
        }
    }
}

```

```

sdi=2;
di=0;
sdbi=0x80;
dbi=0x0001<<(data[di+mode].length-1);
data_end=0;
sdata_end=0;

while (data_end==0) {
    if ((data[di+mode].bin & dbi)==0) {
        sdata[sdi]=sdata[sdi] & (0xFF^sdbi); // data[di].bin[dbi]=0
        cingl=0;
        crc7=(crc & 0x80);
        crc<<=1;
        crc=crc & 0xFE;
        crc=crc | ((crc7>>7)^0x00);
    } else {
        sdata[sdi]=sdata[sdi] | sdbi; // data[di].bin[dbi]=1
        cingl++;
        crc7=(crc & 0x80);
        crc<<=1;
        crc=crc & 0xFE;
        crc=crc | ((crc7>>7)^0x01);
    }
    if (cingl==5) {
        cingl=0;
        if ((sdbi==1) && (sdi!=sdi_max)) {
            sdi++;
            sdbi=0x80;
        } else if ((sdbi==1) && (sdi==sdi_max)) {
            return 0;
        } else {
            sdbi=sdbi>>1;
        }
        sdata[sdi]=sdata[sdi] & (0xFF^sdbi);
    }
    if ((sdbi==1) && (sdi!=sdi_max)) {
        sdi++;
        sdbi=0x80;
    } else if ((sdbi==1) && (sdi==sdi_max)) {
        return 0;
    } else {
        sdbi=sdbi>>1;
    }
    if ((dbi==1) && (di!=5)) {
        di++;
        dbi=0x0001<<(data[di+mode].length-1);
    } else if ((dbi==1) && (di==5)) {
        data_end=1;
    } else {
        dbi=dbi>>1;
    }
}
dbi=0x0080;

```

```

for (i=0;i<8;i++) {
    if ((crc & dbi)==0) {
        sdata[sdi]=sdata[sdi] & (0xFF^sdbi); // crc[dbi]=0
        cingl=0;
    } else {
        sdata[sdi]=sdata[sdi] | sdbi;          // crc[dbi]=1
        cingl++;
    }
    if (cingl==5) {
        cingl=0;
        if ((sdbi==1) && (sdi!=sdi_max)) {
            sdi++;
            sdbi=0x80;
        } else if ((sdbi==1) && (sdi==sdi_max)) {
            return 0;
        } else {
            sdbi=sdbi>>1;
        }
        sdata[sdi]=sdata[sdi] & (0xFF^sdbi);
    }
    if ((sdbi==1) && (sdi!=sdi_max)) {
        sdi++;
        sdbi=0x80;
    } else if ((sdbi==1) && (sdi==sdi_max)) {
        if (i==7) { sdata_end=1; } else { return 0; }
    } else {
        sdbi=sdbi>>1;
    }
    dbi>>=1;
}
while (sdata_end==0) {
    sdata[sdi]=sdata[sdi] & (0xFF^sdbi);
    if ((sdbi==1) && (sdi!=sdi_max)) {
        sdi++;
        sdbi=0x80;
    } else if ((sdbi==1) && (sdi==sdi_max)) {
        sdata_end=1;
    } else {
        sdbi=sdbi>>1;
    }
}
return 1;
}

void view(void) {

int c1,c2;

gotoxy(C1,L5);
printf("SData:");
if (serialize_data()) {
    for (c1=0;c1<=sdi_max;c1++) {
        printbin(sdata[c1],8);
        gotoxy(wherex()+8,wherey());
    }
}
}

```

```

    }
    gotoxy(C1,L6);
    printf("CRC:");
    printbin(crc,8);
    gotoxy(XCur,YCur);
    clreol();
    printf("Press any key to continue...");
    WK;
} else {
    printf(" ERROR: not enough memory in controller to proceed!");
    gotoxy(XCur,YCur);
    clreol();
    printf("Press any key to continue... (no data can be send)");
    WK;
}
}

void send(void) {

    int c1,c2;

    gotoxy(C1,L5);
    printf("SData:");
    if (serialize_data()) {
        for (c1=0;c1<=sdi_max;c1++) {
            printbin(sdata[c1],8);
            gotoxy(wherex()+8,wherey());
        }
        gotoxy(C1,L6);
        printf("CRC:");
        printbin(crc,8);
        // gotoxy(XCur,YCur);
        // clreol();
        // printf("Press any key to send...");
        // WK;

        reset(); //Reset controller
        outportb(donnees,0x00); //Prime=0
        //Send data
        for (c1=0;c1<=sdi_max;c1++) {
            for (c2=0;c2<8;c2++) {
                outportb(donnees,(sdata[c1]<<c2)&(0x80));
                outportb(donnees,inportb(donnees)|(0x40));
            }
        }
        outportb(donnees,0x02);
        delay(1000); //Prime length
        outportb(donnees,0x06); //Enable 300K
        delay(10);
        outportb(donnees,0x04); //Prime=0

        switch (mode) {
            case 0 : last_send="Sel Stim";
                    break;

```

```

        case 6 : last_send="Imp Mes";
                break;
        case 12 : last_send="Perm Stim";
                break;
    }

    gotoxy(XCur,YCur);
    clreol();
    printf("Bits Stream sent");
    delay(TEMPO);
} else {
    printf(" ERROR: not enough memory in controller to proceed!");
//    gotoxy(XCur,YCur);
//    clreol();
//    printf("Press any key to continue... (no data sent)");
//    WK;
}
}

void reset(void) {

    outportb(donnees,inportb(donnees)&0xFB); //EN_Clk300K=0
    outportb(donnees,inportb(donnees)|0x01); //Reset=1
    outportb(donnees,inportb(donnees)&0xFE); //Reset=0

    last_send="None";

    gotoxy(XCur,YCur);
    clreol();
    printf("Controller Reset");
    delay(TEMPO);
}

void prime(void) {

    if ((inportb(donnees)&(0x02))==0) {
        outportb(donnees,inportb(donnees)|0x02);
    } else {
        outportb(donnees,inportb(donnees)&0xFD);
    }
}

void sd_reset(void) {

    int c1,c2;

    sdata[0]=entete;
    sdata[1]=cmd_reset;
    sdata[2]=entete;
    sdata[3]=cmd_reset;
    sdata[4]=entete;
    sdata[5]=cmd_reset;
    sdata[6]=entete;
    sdata[7]=cmd_reset;

```

```

sdata[8]=entete;
sdata[9]=cmd_reset;
sdata[10]=0x00;

reset(); //Reset controller
outportb(donnees,0x00); //prime=0
//Send data
for (c1=0;c1<=sdi_max;c1++) {
    for (c2=0;c2<8;c2++) {
        outportb(donnees,(sdata[c1]<<c2)&(0x80));
        outportb(donnees,inportb(donnees)|(0x40));
    }
}
outportb(donnees,0x02);
delay(1000); //Prime length
outportb(donnees,0x06); //enable 300K
delay(10);
outportb(donnees,0x04); //Prime=0

last_send="Reset";

gotoxy(XCur,YCur);
clreol();
printf("Reset Stream sent");
delay(TEMPO);
}

int rdffloat(float *val, float valini) {

char txt[50];
int pos=0,x,y,c;
int fin=0,choix;

x=wherex(); y=wherey();
*val=valini;
float2txt(val,txt,1);
printf("%s",txt);
gotoxy(x+pos,y);
while (!fin) {
    if (!(choix=getch())) choix=getch()+0x100;
    switch (choix) {
        case 48 :
        case 49 :
        case 50 :
        case 51 :
        case 52 :
        case 53 :
        case 54 :
        case 55 :
        case 56 :
        case 57 : if (!txt[pos]) txt[pos+1]=0;
                    txt[pos]=choix;
                    printf("%c",choix);
                    if (pos<5) pos++;
    }
}

```

```

        gotoxy(x+pos,y);
        break;
    case POINT : if (!txt[pos]) txt[pos+1]=0;
        txt[pos]='.';
        printf(".");
        pos++;
        gotoxy(x+pos,y);
        break;
    case GAUCHE : if (pos) pos--;
        gotoxy(x+pos,y);
        break;
    case DROIT  : if (txt[pos]) pos++;
        gotoxy(x+pos,y);
        break;
    case HOME   : pos=0;
        gotoxy(x+pos,y);
        break;
    case END    : while (txt[pos]) pos++;
        gotoxy(x+pos,y);
        break;
    case ENTER  : fin=1;
        break;
    case ESC    : fin=1;
        *val=valini;
        return 0;
    }
}
if (txt2float(txt,val)) {
    *val=valini;
    return 1;
}
return 0;
}

int txt2float(char *src,float *dest){
    int c=0,point,div;

    *dest=0;
    point=0;
    div=1;
    while (src[c] && c<100) {
        if (src[c]=='.') {
            if (point==1) return 1;
            point=1;
        } else if (!point) {
            *dest*=10.0;
            *dest+=src[c]-48;
        } else {
            div*=10;
            *dest+=(float) (src[c]-48)/(float)div;
        }
        c++;
    }
}

```

```

    if (c==100) return 1;
    return 0;
}

int float2txt(float *src,char *dest,int prec) {

long int c=1,pos=0,decph;
float val;

    val=*src;
    while (val>=10.0) {
        c++;
        val/=10;
    }
    while (c) {
        if ( (val<1.001) && (val>.99) ) {
            val=1.0;
        }
        if ( (val<2.001) && (val>1.99) ) {
            val=2.0;
        }
        if ( (val<3.001) && (val>2.99) ) {
            val=3.0;
        }
        if ( (val<4.001) && (val>3.99) ) {
            val=4.0;
        }
        if ( (val<5.001) && (val>4.99) ) {
            val=5.0;
        }
        if ( (val<6.001) && (val>5.99) ) {
            val=6.0;
        }
        if ( (val<7.001) && (val>6.99) ) {
            val=7.0;
        }
        if ( (val<8.001) && (val>7.99) ) {
            val=8.0;
        }
        if ( (val<9.001) && (val>8.99) ) {
            val=9.0;
        }
        dest[pos]=(char)val+48;
        val-=(float)(dest[pos]-48);
        pos++;
        c--;
        val*=10;
    }
    dest[pos]='.';
    pos++;
    decph=0;
    while (decph<prec) {
        dest[pos]=(char)val+48;
        val-=(float)(dest[pos]-48);

```

```

    pos++;
    val*=10;
    decph++;
}
if (val>=5.0000) dest[pos-1]++;
dest[pos]=0;
return 0;
}

```

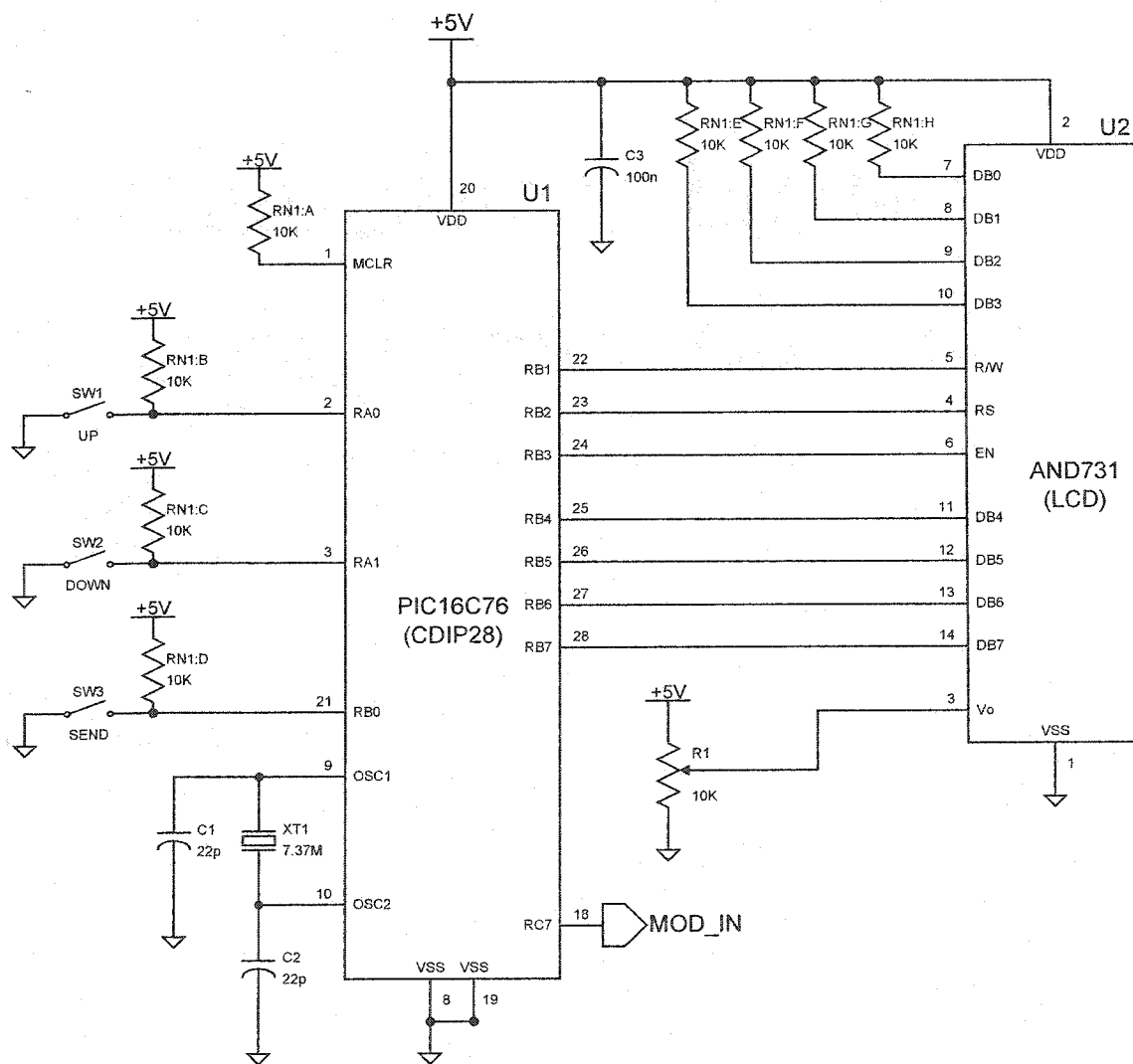


Figure C.5 : Schéma électrique du CPS

```

;*****
;
;   Nom du fichier : Ctrl05.asm
;   Date           : 12 Oct 2000
;   Version        : 1.3
;   Auteur         : E. Schneider
;*****
;   Fichiers requis : P16C76.inc
;*****
;   Notes: Programme assembleur pour le CPS
;*****
;               Configuration par défaut
;
;   PinOut : RA0 = UP | BootUp | 'Z' | in | Pull up
;             RA1 = DOWN | 'Z' | in | Pull up
;             RA2 =      | 'Z' | in | Pull up
;             RA3 =      | 'Z' | in | Pull up
;             RA4 =      | 'Z' | in | Pull up
;             RA5 =      | 'Z' | in | Pull up
;             RB0 = SEND | 'Z' | in | Pull up
;             RB1 = RW   | '0' | out | Pull up
;             RB2 = RS   | '0' | out | Pull up
;             RB3 = EN   | '0' | out | Pull up
;             RB4 = DB4  | 'Z' | in | Pull up
;             RB5 = DB5  | 'Z' | in | Pull up
;             RB6 = DB6  | 'Z' | in | Pull up
;             RB7 = DB7  | 'Z' | in | Pull up
;             RC7-0 = MD | '0' | out
;*****

list      p=16c76           ; Déclaration du processeur
#include   <p16c76.inc>      ; Définition de variables spécifique
du processeur
_CONFIG   _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC
;*****
;               DEFINITION des MACROS
;*****
#define    bank0    bcf STATUS,RP0
#define    bank1    bsf STATUS,RP0

hi macro port,pin
    bsf    port,pin
endm
lo macro port,pin
    bcf    port,pin
endm
movlw macro val,file
    movlw  val
    movwf  file
endm

;*****
;               DEFINITION des CONSTANTES
;*****

```

```

RA      EQU      PORTA
RB      EQU      PORTB
RC      EQU      PORTC

UP      EQU      D'0' ; (RA)
DOWN    EQU      D'1' ; (RA)
RA2     EQU      D'2' ; (RA)
RA3     EQU      D'3' ; (RA)
RA4     EQU      D'4' ; (RA)
RA5     EQU      D'5' ; (RA)

SEND    EQU      D'0' ; (RB)
RW      EQU      D'1' ; (RB)
RS      EQU      D'2' ; (RB)
EN      EQU      D'3' ; (RB)
DB4     EQU      D'4' ; (RB)
DB5     EQU      D'5' ; (RB)
DB6     EQU      D'6' ; (RB)
DB7     EQU      D'7' ; (RB)
MD      EQU      D'7' ; (RB)

KeyUp   EQU      D'0' ; (Flags)
KeyDown EQU      D'1' ; (Flags)
KeySend EQU      D'3' ; (Flags)
RB0IRQ  EQU      D'4' ; (Flags)

INIT1   EQU      B'10101001' ; "0001"
INIT2   EQU      B'10011001' ; "0101"
ENTETE1 EQU      B'10010101' ; "0111"
ENTETE2 EQU      B'01010110' ; "1110"
RESET   EQU      B'10011001' ; "0101"

;*****
;                               DEFINITION des VARIABLES                               *
;*****
Temp0    EQU      0x70 ; for level 0 Sousroutines local use only
Temp1    EQU      0x71 ; for level 1 Sousroutines local use only
Temp2    EQU      0x72 ; for level 2 Sousroutines local use only

ROMindex EQU      0x73
ROMbyte  EQU      0x74
ROMbyteLength EQU 0x75
INDFindex EQU      0x76
NbParam  EQU      0x77
ROMbyteIndex EQU    0x78
FiveOne  EQU      0x79

Menu     EQU      0x7A ; ID of active menu
Flags    EQU      0x7B
RAM      EQU      0x20 ; 0x20 to 0x40 reserved for RAM !WARNING!

;*****
;                               INITIALISATION du VECTEUR de RESET                               *
;*****

```

```

ORG      0x000
        goto    Main

;*****
;                               CODE D'INTERRUPTION MATÉRIELLE
;*****

ORG      0x004
        hi      Flags,RB0IRQ
        return

;*****
;***** PROGRAMME PRINCIPAL *****
;*****

Main
; Initialisation des ports
        clrf    RA
        clrf    RB
        clrf    RC
        bank1
        hi      ADCON1,PCFG2
        hi      ADCON1,PCFG1
        movlwf  B'11110001',RB
        clrf    RC
        bank0

; Initialisation des variables globales
        movlwf  B'00100000',Menu
        movf    RA,W
        andlw   B'00000011'
        movwf   Flags
        btfsc   RB,SEND
        hi      Flags,KeySend

; Petit temps mort pour attendre le power up du LCD
        movlw   B'00000110'
        call    Standby

; Initialisation du LCD
        movlw   B'00000010' ; Init LCD in 4 bits commande
        call    SendCmd
        movlw   B'00101000' ; Fonction Set
        call    SendCmd
        movlw   B'00001100' ; Display On
        call    SendCmd
        movlw   B'00000001' ; Display Clear
        call    SendCmd
        movlw   B'00000110' ; Entry Mode Set
        call    SendCmd
        call    DisplayMenu
        call    SerializeData

; Programme principal
        bank1
        btfss   Flags,KeySend
        lo      OPTION_REG,INTEDG
        bank0

KeyUpTest
        clrwdt

```

```

                btfss    Flags,KeyUp
                goto     KeyUpTest1
KeyUpTest0
                btfsc    RA,UP
                goto     KeyDownTest
                call     MenuUp
KeyUpWait0
                clrwdt
                btfss    RA,UP
                goto     KeyUpWait0
                goto     KeyUpDebounce
KeyUpTest1
                btfss    RA,UP
                goto     KeyDownTest
                call     MenuUp
KeyUpWait1
                clrwdt
                btfsc    RA,UP
                goto     KeyUpWait1
KeyUpDebounce
                movlw    B'00000010'
                call     Standby

KeyDownTest
                btfss    Flags,KeyDown
                goto     KeyDownTest1
KeyDownTest0
                btfsc    RA,DOWN
                goto     KeySendTest
                call     MenuDown
KeyDownWait0
                clrwdt
                btfss    RA,DOWN
                goto     KeyDownWait0
                goto     KeyDownDebounce
KeyDownTest1
                btfss    RA,DOWN
                goto     KeySendTest
                call     MenuDown
KeyDownWait1
                clrwdt
                btfsc    RA,DOWN
                goto     KeyDownWait1
KeyDownDebounce
                movlw    B'00000010'
                call     Standby
KeySendTest
                btfss    Flags,KeySend
                goto     KeySendTest1
KeySendTest0
                btfsc    RB,SEND
                goto     KeyUpTest
                call     SendData
                goto     KeyUpTest

```

```

KeySendTest1
    btfss    RB,SEND
    goto     KeyUpTest
    call     SendData
    goto     KeyUpTest

;*****
;***** FIN du PROGRAMME PRINCIPAL *****
;*****

;*****
;***** SOUSROUTINES *****
;*****

;*****
;***** MENUUP SUBROUTINE [LEVEL 2] *****
; Action    : Affiche le menu suivant
; Entrées   : N/A
; Registres: Menu
; Sousroutines: DisplayMenu, SerializeData, Standby
;*****
MenuUp
    movlw    D'4'
    addwf    Menu,F
    movlw    B'01101000'
    xorwf    Menu,W
    bnz      MenuUpRefresh
    movlwf   B'00100000',Menu

MenuUpRefresh
    call     DisplayMenu
    call     SerializeData
    movlw    B'00000010'
    call     Standby
    return

;*****
;***** MENUDOWN SUBROUTINE [LEVEL 2] *****
; Action    : Affiche le menu précédant
; Entrées   : N/A
; Registres: Menu
; Sousroutines: DisplayMenu, SerializeData, Standby
;*****
MenuDown
    movlw    D'4'
    subwf    Menu,F
    movlw    B'00011100'
    xorwf    Menu,W
    bnz      MenuDownRefresh
    movlwf   B'01100100',Menu

MenuDownRefresh
    call     DisplayMenu
    call     SerializeData
    movlw    B'00000010'
    call     Standby

```

return

```

;*****
;                               SENDDATA SUBROUTINE [LEVEL 2]
;                               *
; Action   : Transmet les données en RAM au modulateur
; Entrées  : N/A
; Registres: INTCON, FSR, Temp2
; Sousroutines: DisplayMenu, Standby
;*****
SendData

```

```

    lo      Flags,RB0IRQ
    movlwf  B'10010000',INTCON    ; Enable interrupt on RB0
    movlw   B'00000100'
    call    Standby
    btfsc   Flags,RB0IRQ
    goto    EndEndSendData
    movf    Menu,W
    movwf   Temp2                ;Save Menu in Temp2
    movlwf  0x68,Menu
    call    DisplayMenu          ;Display Message "Sending Data"
    hi      RC,MD
    movlw   B'00000100'
    call    Standby              ; Prime for ~1s
    btfsc   Flags,RB0IRQ
    goto    EndSendData
    movlw   RAM
    movwf   FSR
    movfw   INDF

```

LoopSendData

```

    movwf   RC                  ; 1 cycle
    incf    FSR,F               ; 1 cycle
    nop                                           ; 1 cycle

    rlf     RC,F                ; 1 cycle
    btfsc   FSR,6               ; 1 cycle ou 2 cycle
    movlw   RAM+5               ; 1 cycle ou 0 cycle

    rlf     RC,F                ; 1 cycle
    btfsc   FSR,6               ; 1 cycle ou 2 cycle
    movwf   FSR                 ; 1 cycle ou 0 cycle

    rlf     RC,F                ; 1 cycle
    movfw   INDF                ; 1 cycle
    clrwdt                      ; 1 cycle

    rlf     RC,F                ; 1 cycle
    btfsc   Flags,RB0IRQ; 1 cycle ou 2 cycle
    goto    EndSendData ; 2 cycle ou 0 cycle

    rlf     RC,F                ; 1 cycle
    btfsc   Flags,RB0IRQ; 1 cycle ou 2 cycle
    goto    EndSendData ; 2 cycle ou 0 cycle

    rlf     RC,F                ; 1 cycle

```

```

        btfsc    Flags,RB0IRQ; 1 cycle ou 2 cycle
        goto     EndSendData ; 2 cycle ou 0 cycle

        rlf      RC,F          ; 1 cycle
        goto     LoopSendData ; 2 cycle
EndSendData
        clrf     INTCON
        clrf     RC
        movlwf   0x6C,Menu
        call     DisplayMenu    ; Display message "Please wait"
        movlw    B'00000110'
        call     Standby
        movf     Temp2,W
        movwf    Menu          ; Restore Menu
        call     DisplayMenu
EndEndSendData
        return   ; of SendData subroutine

```

```

;*****
;                               DISPLAYMENU SUBROUTINE [LEVEL 1]      *
; Action   : Affiche le menu courant sur le LCD                      *
; Entrées  : N/A                                                    *
; Registres: ROMindex, Temp1                                         *
; Sousroutines: SendCmd, GetDataROM, SendChar                       *
;*****

```

DisplayMenu

```

        movlw    B'00000001' ; Clear display
        call     SendCmd      ;
        clrf     ROMindex
        clrf     Temp1

```

LoopDisplayMenu

```

        movfw    ROMindex
        call     GetDataROM
        call     SendChar
        incf     ROMindex,F
        incf     Temp1,F
        btfss    Temp1,4      ; test if Temp1 = 16
        goto     LoopDisplayMenu
        btfsc    ROMindex,5    ; test if ROMindex < 32
        goto     EndDisplayMenu
        movlw    B'11000000'   ; Go to second line of LCD
        call     SendCmd      ;
        clrf     Temp1
        goto     LoopDisplayMenu

```

EndDisplayMenu

```

        return   ; of DisplayMenu subroutine

```

```

;*****
;                               SERIALIZEDATA SUBROUTINE [LEVEL 1]    *
; Action   : Prepare les données en RAM au format Manchester        *
; Entrées  : N/A                                                    *
; Registres: ROMindex, ROMbyte, ROMbyteLength, INDFindex, NbParam  *
;           ROMbyteIndex, FiveOne                                   *
; Sousroutines: GetDataROM                                          *
;*****

```

```

;*****
SerializeData
    movlwf  RAM,FSR
    movlw   B'10101010'

LoopSerialize
    movwf   INDF
    incf    FSR,F
    btfss   FSR,6
    goto    LoopSerialize
    movwf   INDF

    movlwf  RAM,FSR
    movlwf  INIT1,INDF
    incf    FSR,F
    movlwf  INIT2,INDF
    incf    FSR,F
    movlwf  ENTETE1,INDF
    incf    FSR,F
    movlwf  ENTETE2,INDF
    incf    FSR,F
    movlwf  RESET,INDF
    incf    FSR,F

    movlwf  INIT1,INDF
    incf    FSR,F
    movlwf  INIT2,INDF
    incf    FSR,F
    movlwf  ENTETE1,INDF
    incf    FSR,F
    movlwf  ENTETE2,INDF
    incf    FSR,F

    movlwf  D'8',NbParam
    movlwf  D'4',INDFindex
    movlwf  D'32',ROMindex
    clrf    FiveOne

Loop1Serialize
    movfw   ROMindex
    call    GetDataROM
    movwf   ROMbyteLength

Loop2Serialize
    movlwf  D'8',ROMbyteIndex
    incf    ROMindex,F
    movfw   ROMindex
    call    GetDataROM
    movwf   ROMbyte

Loop3Serialize
    btfsc   ROMbyte,7
    goto    OneManchester

ZeroManchester
    clrf    FiveOne
    hi      INDF,1 ; Zero Manchester
    lo      INDF,0 ; Zero Manchester
    goto    ContinuelSerialize

```

```

OneManchester
    incf    FiveOne,F
    lo      INDF,1  ; One Manchester
    hi      INDF,0  ; One Manchester

Continue1Serialize
    decfsz  INDFindex,F
    goto    Continue2Serialize
    movlw   D'4',INDFindex
    incf    FSR,F
    goto    Continue3Serialize

Continue2Serialize
    rlf     INDF,F
    rlf     INDF,F

Continue3Serialize
    movlw   D'5'
    subwf   FiveOne,W
    bz      ZeroManchester
    decfsz  ROMbyteLength,F
    goto    Continue4Serialize
    incf    ROMindex,F
    decfsz  NbParam,F
    goto    Loop1Serialize
    goto    FinishSerialize

Continue4Serialize
    decfsz  ROMbyteIndex,F
    goto    Continue5Serialize
    goto    Loop2Serialize

Continue5Serialize
    rlf     ROMbyte,F
    goto    Loop3Serialize

FinishSerialize
    hi      INDF,1  ; Zero Manchester
    lo      INDF,0  ; Zero Manchester
    decf    INDFindex,F
    bz      EndSerializedData
    rlf     INDF,F
    rlf     INDF,F
    goto    FinishSerialize

EndSerializedData
    return ; of serialize subroutine

```

```

;*****
;                               STANDBY SUBROUTINE [LEVEL 0]                *
; Action   : Force le CPU en mode standby pendant X secondes             *
; Entrées  : W[2..0]=WDT prescaler bits                                   *
; Registres: OPTION_REG, INTCON                                           *
;*****
Standby

```

```

    bank1
    iorwf   OPTION_REG,F
    addlw   B'11111000'
    andwf   OPTION_REG,F
    hi      OPTION_REG,PSA
    movlw   B'11111000'

```

```

        andwf    INTCON,F
        clrwdt
        nop
        sleep           ; force CPU in standby
        nop
        movlw     B'00000111'
        iorwf     OPTION_REG,F      ; prog WDT for max delay
        bank0

EndStandby
        return ; of Standby subroutine

;*****
;                               SENDCHAR SUBROUTINE [LEVEL 0]
;                               *
; Action   : Transmet un caractère au LCD en mode 4 bits
; Entrées  : W = Code ASCII du caractère à transmettre
; Registres: Temp0
;*****
SendChar
        movwf     Temp0
        hi        RB,RS ; Selecte LCD data register
        lo        RB,RW ; Write to LCD register
        nop
        hi        RB,EN ; force EN=1
        bank1
        movlw     B'00000001',RB      ; Config RB[7..0] as output
        bank0
        movfw     Temp0
        andlw     B'11110000'
        iorlw     B'00001100'
        movwf     RB ; Send first (Most significant) nibble to LCD
        nop
        lo        RB,EN ; Enable transfer to LCD (EN=0)
        swapf     Temp0,W
        andlw     B'11110000'
        iorlw     B'00001100'
        movwf     RB ; Send second (Less significant) nibble to LCD

        (EN=1)
        nop
        lo        RB,EN ; Enable transfer to LCD (EN=0)
        bank1
        movlw     B'11110001',RB ; Config RB[7..4] as input
        bank0
        lo        RB,RS ; Selecte LCD instruction register
        hi        RB,RW ; Read from LCD register
        clrf      Temp0

LoopSendChar
        incf      Temp0,F
        lo        RB,EN
        nop
        nop
        hi        RB,EN
        nop
        nop
        btfsc     Temp0,0

```

```

        btfsc    RB,DB7 ; Test if LCD still busy
        goto     LoopSendChar

EndSendChar
        clrf     RB
        return   ; of SendChar subroutine

;*****
;                               SENDCMD SUBROUTINE [LEVEL 0]
;                               *
; Action      : Transmet une commande au LCD en mode 4 bits
;               *
; Entrées    : W = Code de la commande à transmettre
;               *
; Registres: Temp0
;               *
;*****
SendCmd
        movwf    Temp0
        lo       RB,RS ; Selecte LCD instruction register
        lo       RB,RW ; Write to LCD register
        nop
        hi       RB,EN ; force EN=1
        bank1
        movlwf   B'00000001',RB ; Config RB[7..0] as output
        bank0
        movfw    Temp0
        andlw    B'11110000'
        iorlw    B'00001000'
        movwf    RB ; Send first (Most significant) nibble to LCD
        nop
        lo       RB,EN ; Enable transfer to LCD (EN=0)
        swapf    Temp0,W
        andlw    B'11110000'
        iorlw    B'00001000'
        movwf    RB ; Send second (Less significant) nibble to LCD

        (EN=1)
        nop
        lo       RB,EN ; Enable transfer to LCD (EN=0)
        bank1
        movlwf   B'11110001',RB ; Config RB[7..4] as input
        bank0
        hi       RB,RW ; Read LCD register
        clrf     Temp0

LoopSendCmd
        incf     Temp0,F
        lo       RB,EN
        nop
        nop
        hi       RB,EN
        nop
        nop
        btfsc    Temp0,0
        btfsc    RB,DB7 ; Test if LCD still busy
        goto     LoopSendCmd

EndSendCmd
        clrf     RB
        return   ; of SendCmd subroutine

```

```

;*****
;                               GETDATAROM SUBROUTINE [Level 0]
; Action   : Passerelle pour la lecture de données en ROM
; Entrées  : W = offset en ROM du segment de 64 bits à adresser
; Registres: Temp0, PCLATH, Menu, PCL
;*****
GetDataROM

```

```

    movwf    Temp0
    swapf    Menu,W
    movwf    PCLATH
    andlw    B'11000000'
    addwf    Temp0,W
    movwf    PCL

```

```
EndGetDataROM
```

```
; no "return" because of DT definition of Data in DataRom
```

```

;*****
;***** DONNÉES EN ROM *****
;*****

```

```

ORG     0x200
    DT      " STIM#1 LF ONLY "
    DT      ".9mA 175us 30Hz "
    DT      D'8',B'10011001' ; CMD      (8 bits)
    DT      D'8',B'01110010' ; LFA=.9mA (8 bits)
    DT      D'8',B'00000000' ; HFA=0mA  (8 bits)
    DT      D'6',B'11010000' ; LFW=175us (6 bits + "00" )
    DT      D'6',B'00000000' ; HFW=0us  (6 bits + "00" )
    DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")
    DT      D'10',B'00000000',B'00000000' ; HFP=0Hz  (8 bits + 2
bits + "000000")
    DT      D'8',B'01110011' ; CRC      (8 bits)
ORG     0x240
    DT      " STIM#2 HF BLOCK"
    DT      ".9m/175 1m/60 "
    DT      D'8',B'10011001' ; CMD      (8 bits)
    DT      D'8',B'01110010' ; LFA=.9mA (8 bits)
    DT      D'8',B'01111111' ; HFA=1mA  (8 bits)
    DT      D'6',B'11010000' ; LFW=175us (6 bits + "00" )
    DT      D'6',B'01000100' ; HFW=60us (6 bits + "00" )
    DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")
    DT      D'10',B'01111101',B'00000000' ; HFP=600Hz(8 bits + 2
bits + "000000")
    DT      D'8',B'01100000' ; CRC      (8 bits)
ORG     0x280
    DT      " STIM#3 HF BLOCK"
    DT      ".9m/175 1.5m/60 "
    DT      D'8',B'10011001' ; CMD      (8 bits)
    DT      D'8',B'01110010' ; LFA=.9mA (8 bits)
    DT      D'8',B'10111111' ; HFA=1.5mA (8 bits)
    DT      D'6',B'11010000' ; LFW=175us (6 bits + "00" )
    DT      D'6',B'01000100' ; HFW=60us (6 bits + "00" )
    DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")

```

```

DT      D'10',B'011111101',B'000000000' ; HFP=600Hz(8 bits + 2
bits + "000000")
DT      D'8',B'011011100' ; CRC          (8 bits)
ORG      0x2C0
DT      " STIM#4 HF BLOCK"
DT      ".9m/175 1m/100 "
DT      D'8',B'10011001' ; CMD          (8 bits)
DT      D'8',B'01110010' ; LFA=.9mA    (8 bits)
DT      D'8',B'01111111' ; HFA=1mA     (8 bits)
DT      D'6',B'11010000' ; LFW=175us   (6 bits + "00" )
DT      D'6',B'01110100' ; HFW=100us   (6 bits + "00" )
DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")
DT      D'10',B'011111101',B'000000000' ; HFP=600Hz(8 bits + 2
bits + "000000")
DT      D'8',B'011011100' ; CRC          (8 bits)
ORG      0x300
DT      " STIM#5 HF BLOCK"
DT      ".9m/175 1.5m/100"
DT      D'8',B'10011001' ; CMD          (8 bits)
DT      D'8',B'01110010' ; LFA=.9mA    (8 bits)
DT      D'8',B'10111111' ; HFA=1.5mA   (8 bits)
DT      D'6',B'11010000' ; LFW=175us   (6 bits + "00" )
DT      D'6',B'01110100' ; HFW=100us   (6 bits + "00" )
DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")
DT      D'10',B'011111101',B'000000000' ; HFP=600Hz(8 bits + 2
bits + "000000")
DT      D'8',B'011000000' ; CRC          (8 bits)
ORG      0x340
DT      " STIM#6 HF BLOCK"
DT      "1.1m/175 1.5/100"
DT      D'8',B'10011001' ; CMD          (8 bits)
DT      D'8',B'10001100' ; LFA=1.1mA   (8 bits)
DT      D'8',B'10111111' ; HFA=1.5mA   (8 bits)
DT      D'6',B'11010000' ; LFW=175us   (6 bits + "00" )
DT      D'6',B'01110100' ; HFW=100us   (6 bits + "00" )
DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")
DT      D'10',B'011111101',B'000000000' ; HFP=600Hz(8 bits + 2
bits + "000000")
DT      D'8',B'10001111' ; CRC          (8 bits)
ORG      0x380
DT      " STIM#7 LF ONLY "
DT      "1.5mA 210us 30Hz"
DT      D'8',B'10011001' ; CMD          (8 bits)
DT      D'8',B'10111111' ; LFA=1.5mA   (8 bits)
DT      D'8',B'00000000' ; HFA=0mA     (8 bits)
DT      D'6',B'11111100' ; LFW=210us   (6 bits + "00" )
DT      D'6',B'00000000' ; HFW=0us     (6 bits + "00" )
DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")
DT      D'10',B'00000000',B'000000000' ; HFP=0Hz (8 bits + 2
bits + "000000")

```

```

DT      D'8',B'01101101' ; CRC      (8 bits)
ORG     0x3C0
DT      " STIM#8 LF ONLY "
DT      " 2mA 175us 30Hz "
DT      D'8',B'10011001' ; CMD      (8 bits)
DT      D'8',B'11111111' ; LFA=2mA (8 bits)
DT      D'8',B'00000000' ; HFA=0mA  (8 bits)
DT      D'6',B'11010000' ; LFW=175us (6 bits + "00" )
DT      D'6',B'00000000' ; HFW=0us  (6 bits + "00" )
DT      D'14',B'10011000',B'01011000' ; LFP=30Hz (8 bits + 6
bits + "00")
DT      D'10',B'00000000',B'00000000' ; HFP=0Hz  (8 bits + 2
bits + "000000")
DT      D'8',B'10101011' ; CRC      (8 bits)
ORG     0x400
DT      " Permanent Stim "
DT      "      Turn ON      "
DT      D'8',B'10101010' ; CMD
DT      D'8',B'01110111' ; CMD_PIC
DT      D'8',B'00000001' ; Amp=NA
DT      D'8',B'00000001' ; PulseWide=NA
DT      D'8',B'00000001' ; Frequency=NA
DT      D'8',B'00000001' ; TimeOnOff=NA
DT      D'12',B'01111011',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'01101010' ; CRC
ORG     0x440
DT      " Permanent Stim "
DT      "      Turn OFF     "
DT      D'8',B'10101010' ; CMD
DT      D'8',B'11011101' ; CMD_PIC
DT      D'8',B'00000001' ; Amp=NA
DT      D'8',B'00000001' ; PulseWide=NA
DT      D'8',B'00000001' ; Frequency=NA
DT      D'8',B'00000001' ; TimeOnOff=NA
DT      D'12',B'11100001',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'01101001' ; CRC
ORG     0x480
DT      " PROG#1 18s/10s "
DT      "30uA 150us 20Hz "
DT      D'8',B'10101010' ; CMD
DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00000100' ; Amp=30uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s
DT      D'12',B'10001100',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000011' ; CRC
ORG     0x4C0
DT      " PROG#2 18s/10s "
DT      "50uA 150us 20Hz "
DT      D'8',B'10101010' ; CMD

```

```

DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00000111' ; Amp=50uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s
DT      D'12',B'10001111',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000011' ; CRC
ORG      0x500
DT      " PROG#3 18s/10s "
DT      "70uA 150us 20Hz "
DT      D'8',B'10101010' ; CMD
DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00001001' ; Amp=70uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s
DT      D'12',B'10010001',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000010' ; CRC
ORG      0x540
DT      " PROG#4 18s/10s "
DT      "100uA 150us 20Hz"
DT      D'8',B'10101010' ; CMD
DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00001100' ; Amp=100uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s
DT      D'12',B'10010100',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000010' ; CRC
ORG      0x580
DT      " PROG#5 18s/10s "
DT      "150uA 150us 20Hz"
DT      D'8',B'10101010' ; CMD
DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00010011' ; Amp=150uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s
DT      D'12',B'10011011',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000011' ; CRC
ORG      0x5C0
DT      " PROG#6 18s/10s "
DT      "200uA 150us 20Hz"
DT      D'8',B'10101010' ; CMD
DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00011001' ; Amp=200uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s

```

```

DT      D'12',B'10100001',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000000' ; CRC
ORG      0x600
DT      " PROG#7 18s/10s "
DT      "300uA 150us 20Hz"
DT      D'8',B'10101010' ; CMD
DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00100110' ; Amp=300uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s
DT      D'12',B'10101110',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000011' ; CRC
ORG      0x640
DT      " PROG#8 18s/10s "
DT      "400uA 150us 20Hz"
DT      D'8',B'10101010' ; CMD
DT      D'8',B'10111011' ; CMD_PIC
DT      D'8',B'00110011' ; Amp=400uA
DT      D'8',B'00010010' ; PulseWide=150us
DT      D'8',B'10000000' ; Frequency=20Hz
DT      D'8',B'00111011' ; TimeOnOff=18s/10s
DT      D'12',B'10111011',B'00000000' ; Checksum (8 bits +
"0000")
DT      D'8',B'00000011' ; CRC
ORG      0x680
DT      ">>  SENDING  <<"
DT      ">>  DATA    <<"
ORG      0x6C0
DT      "!!  PLEASE  !!"
DT      "!!  WAIT    !!"
END ; of program code.

```

ANNEXE D

SCHÉMAS DU MODULE D’AFFICHAGE DE L’IMPÉDANCE ET DU MODULE DE STABILISATION DE LA PORTEUSE

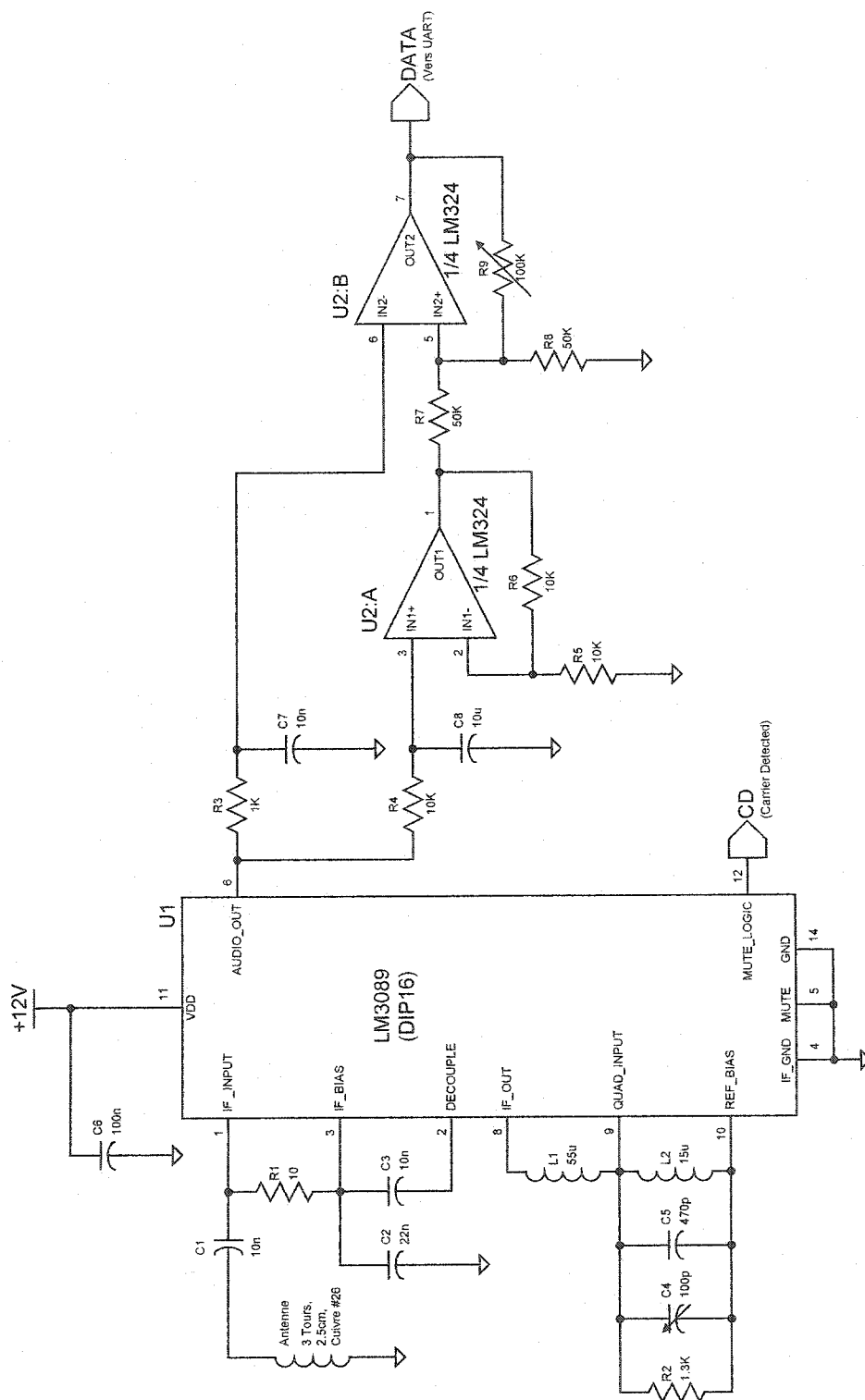


Figure D.1 : Schéma électrique du module d'affichage d'impédance

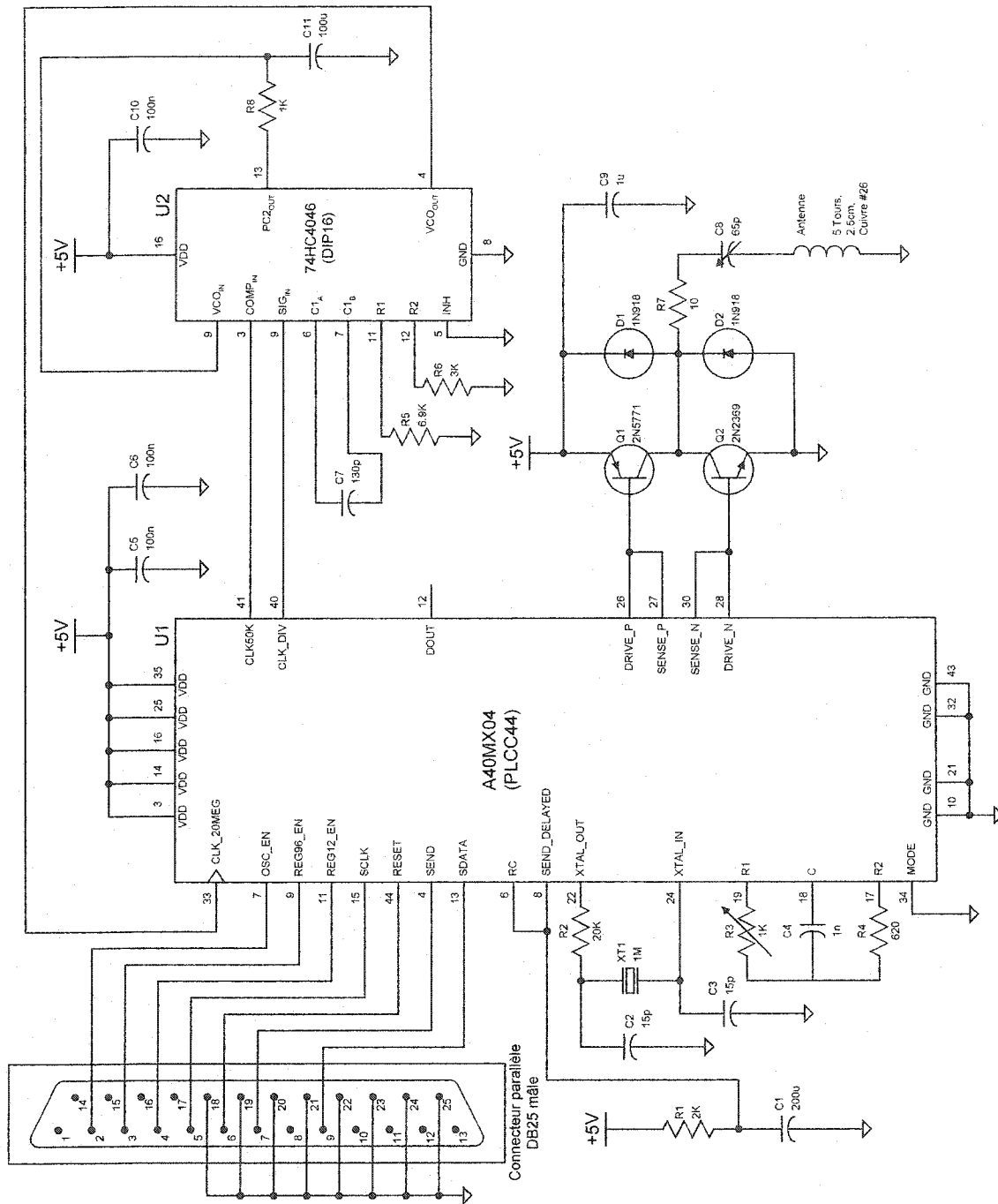


Figure D.2 : Schéma électrique du module de stabilisation de la porteuse

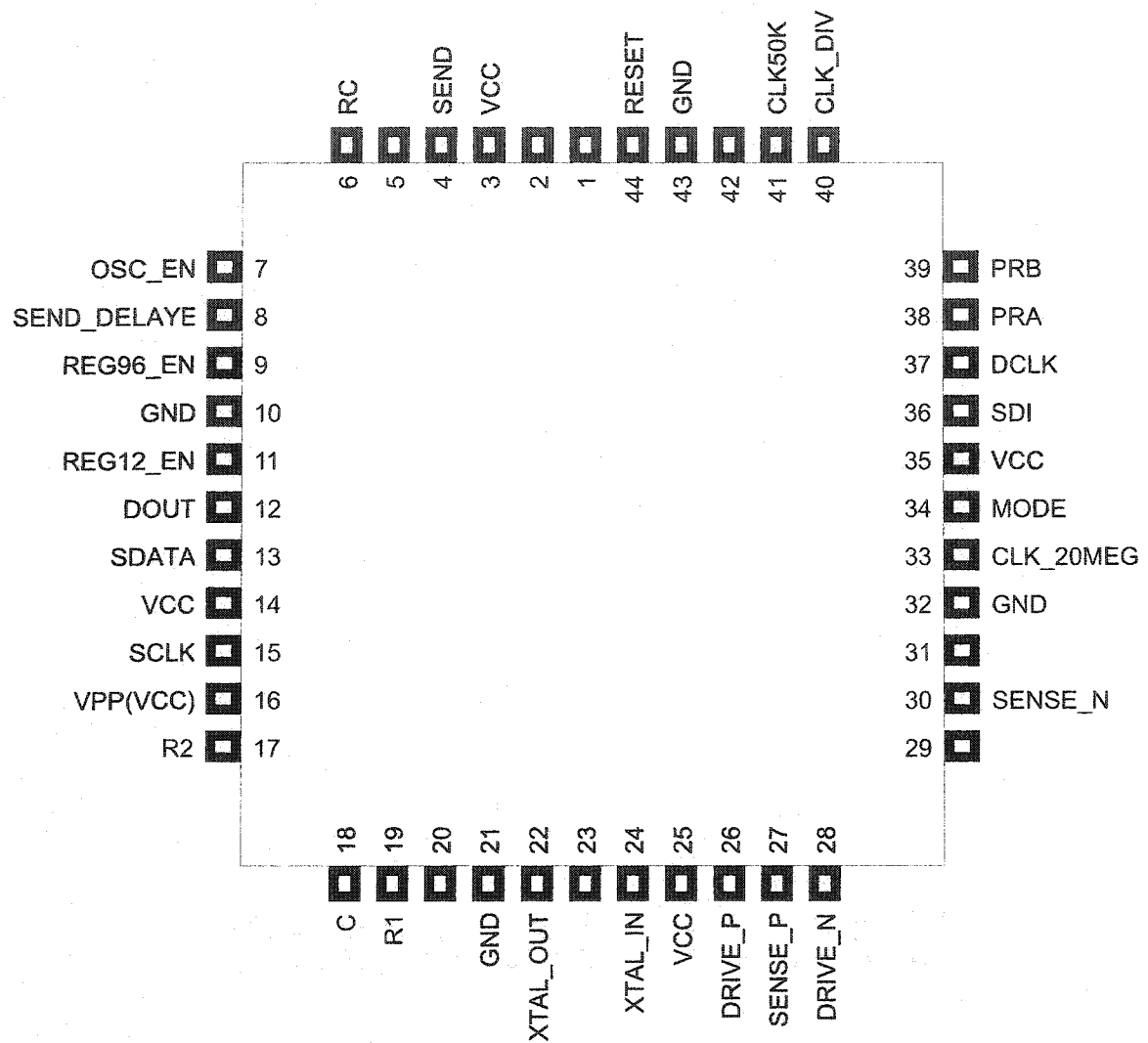


Figure D.3 : Répartition des broches du FPGA du module de stabilisation de la porteuse

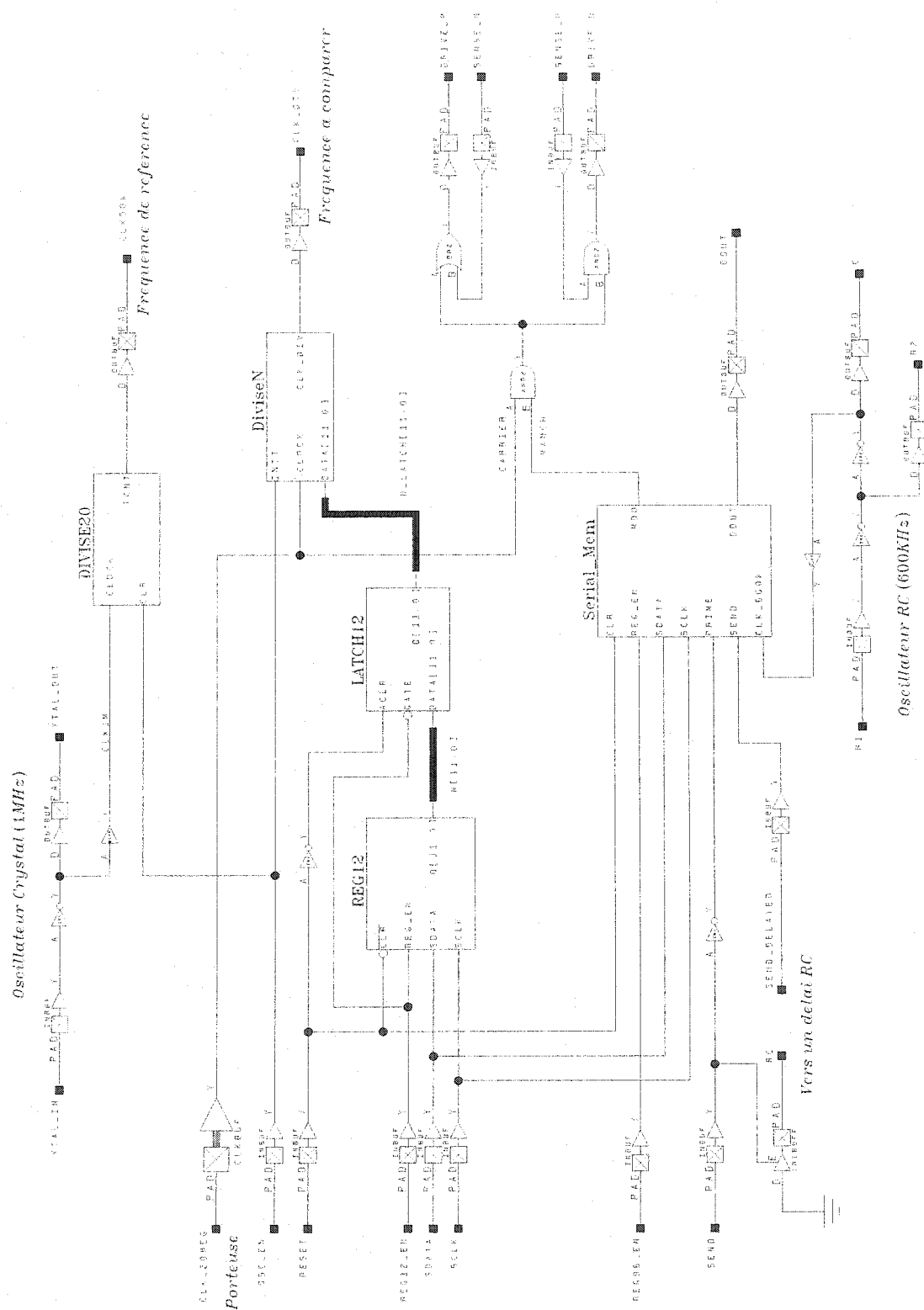


Figure D.4 : Schéma logique de niveau hiérarchique supérieur du FPGA du module de stabilisation de la porteuse

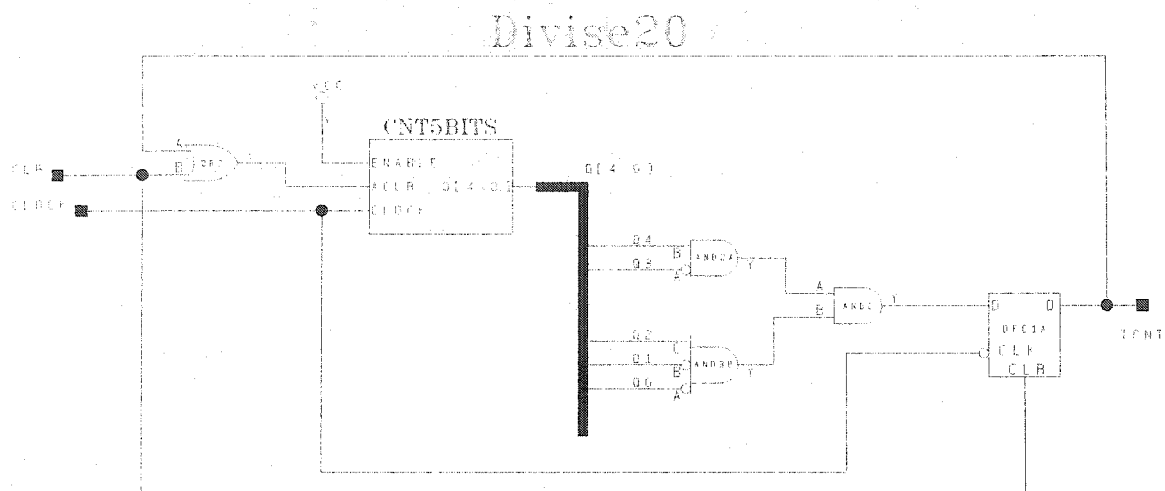


Figure D.5 : Schéma logique du diviseur de fréquence par 20 du FPGA du module de stabilisation de la porteuse

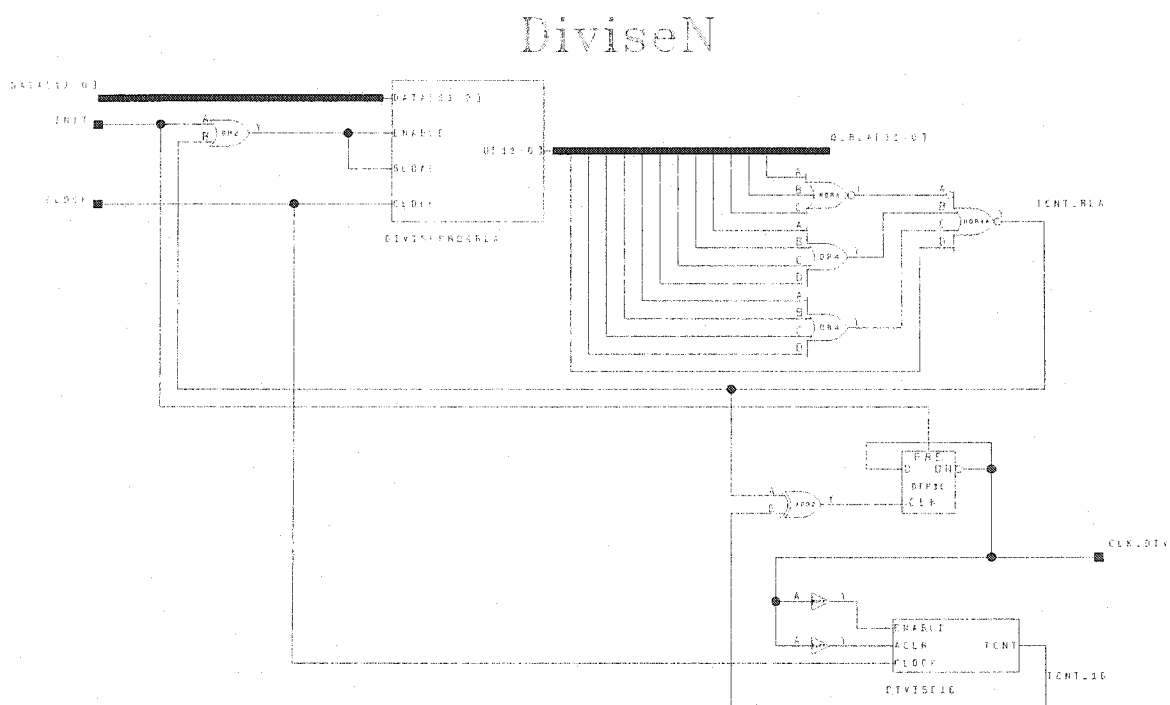


Figure D.6 : Schéma logique du diviseur de fréquence programmable du FPGA du module de stabilisation de la porteuse

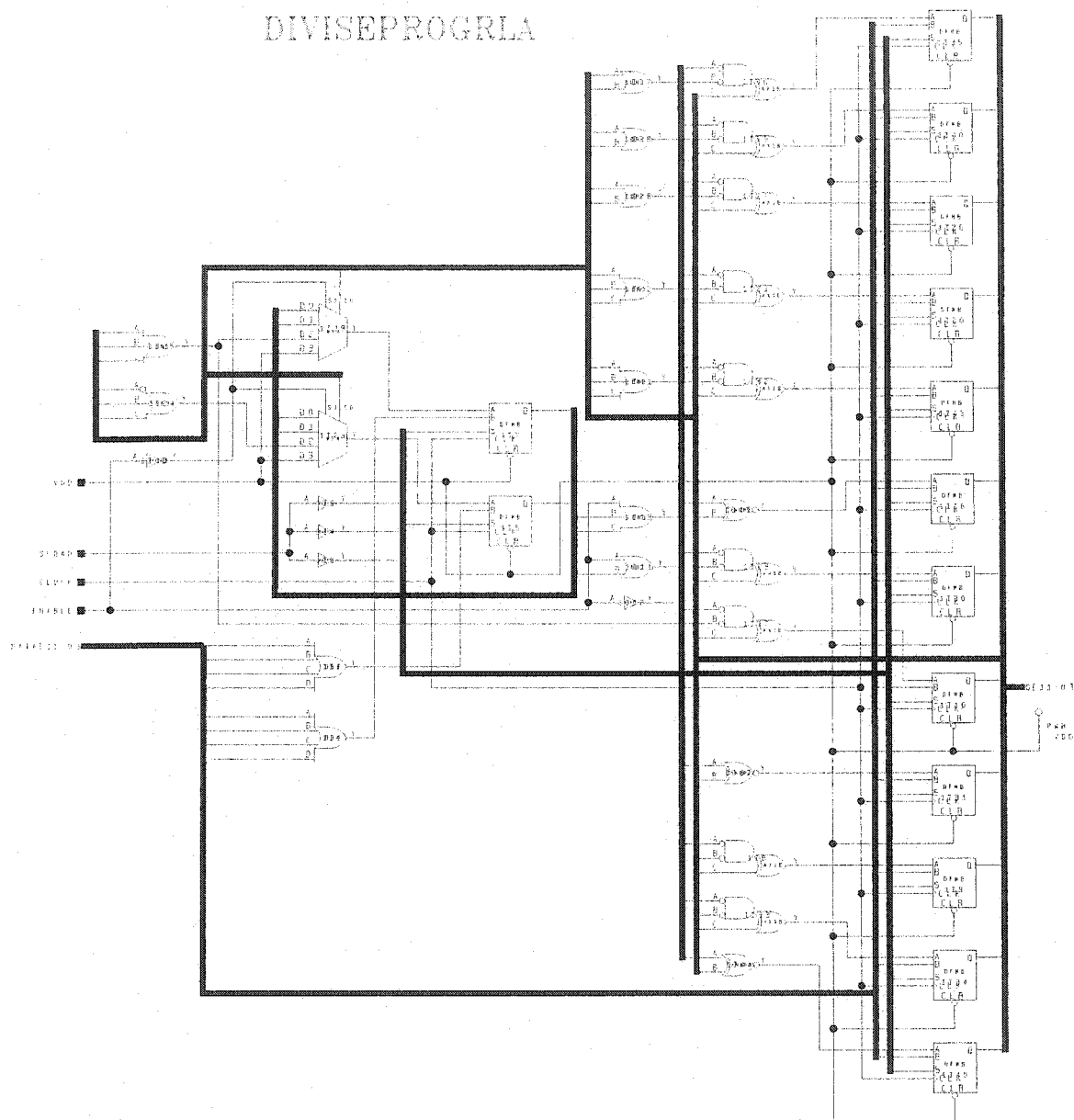


Figure D.7 : Schéma logique du compteur programmable synchrone du FPGA du module de stabilisation de la porteuse

Reg12

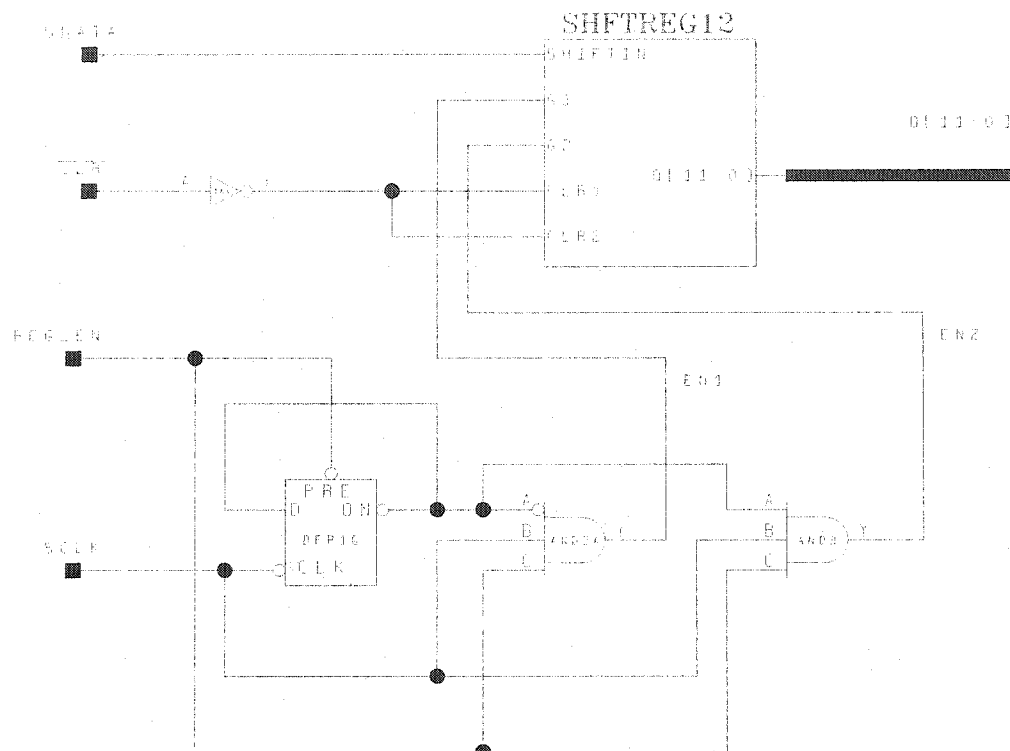


Figure D.8 : Schéma logique du contrôleur de registre à décalage de 12 bits du FPGA du module de stabilisation de la porteuse

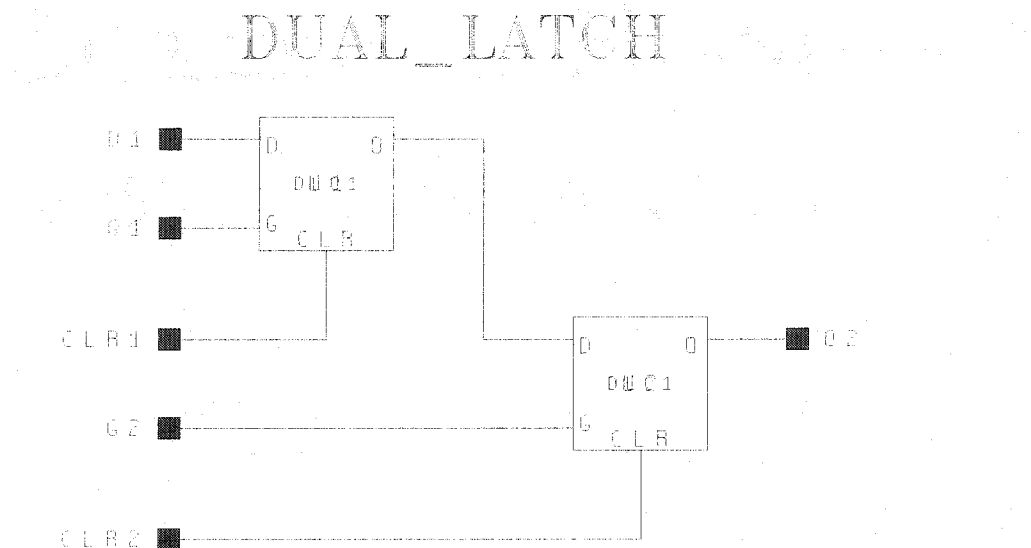


Figure D.9 : Schéma logique d'un registre 1 bit à double latch du FPGA du module de stabilisation de la porteuse

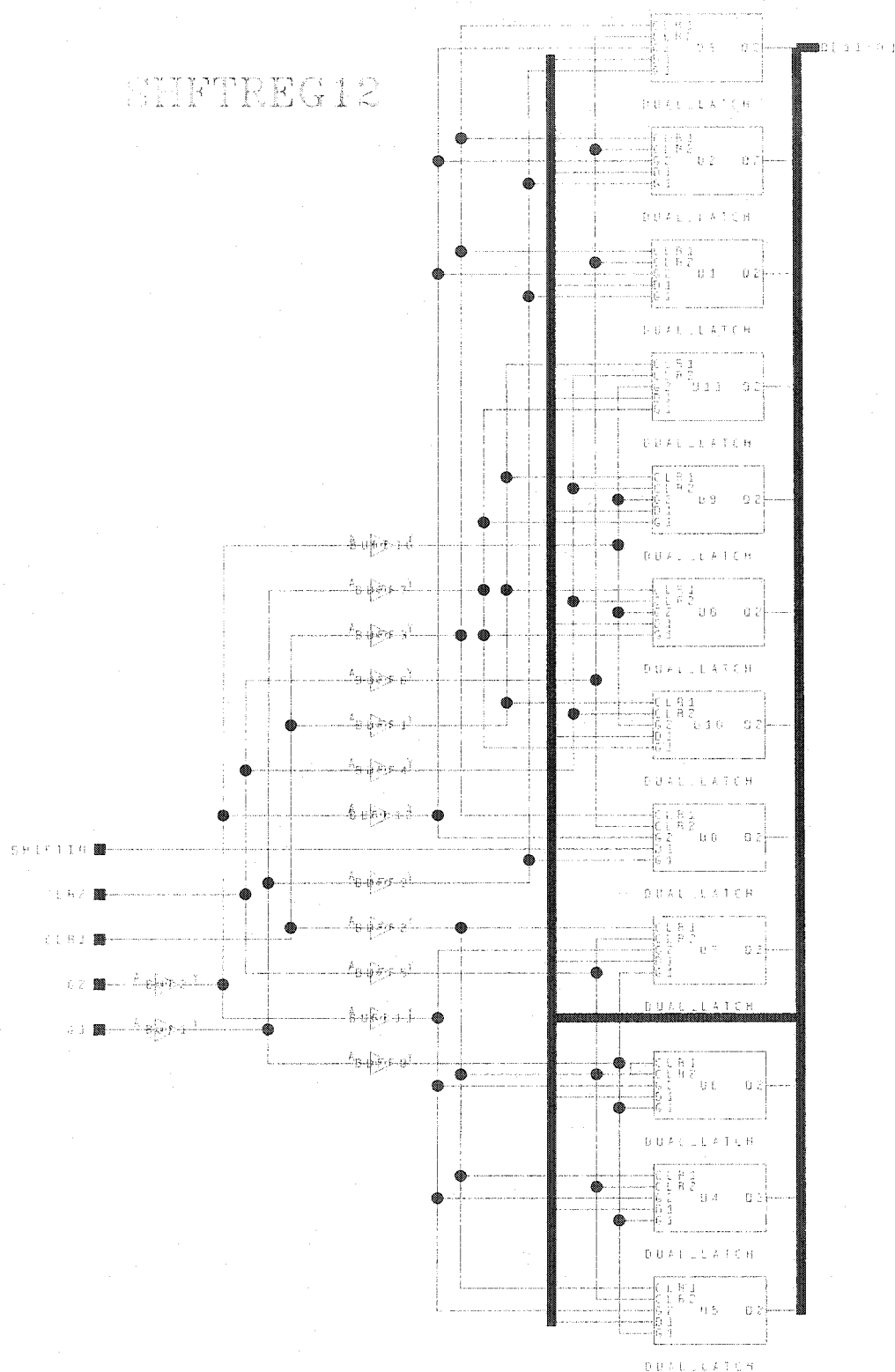
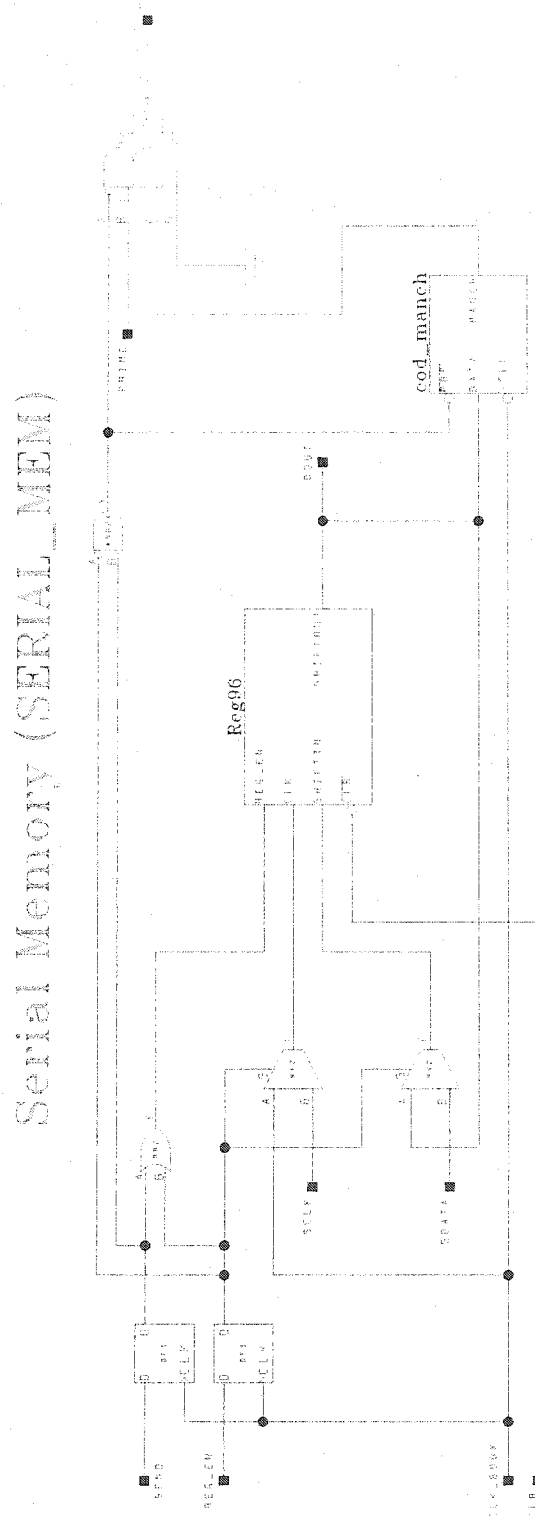


Figure D.10 : Schéma logique du registre à décalage de 12 bits du FPGA du module de stabilisation de la porteuse



Reg96

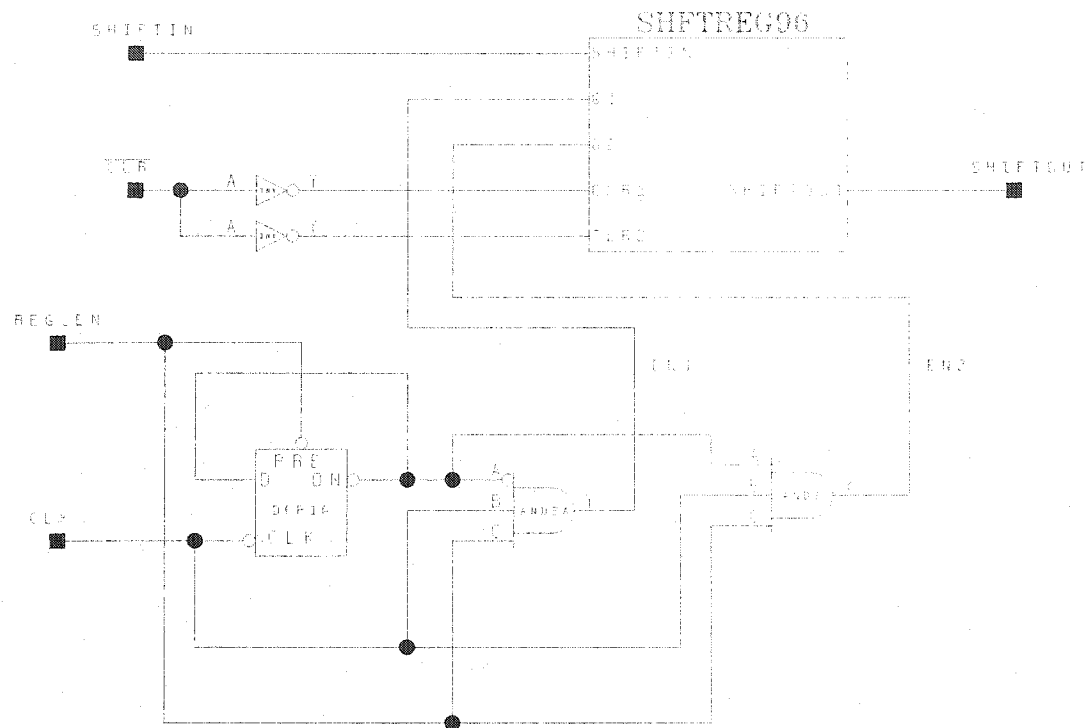


Figure D.12 : Schéma logique du contrôleur de registre à décalage de 96 bits du FPGA du module de stabilisation de la porteuse

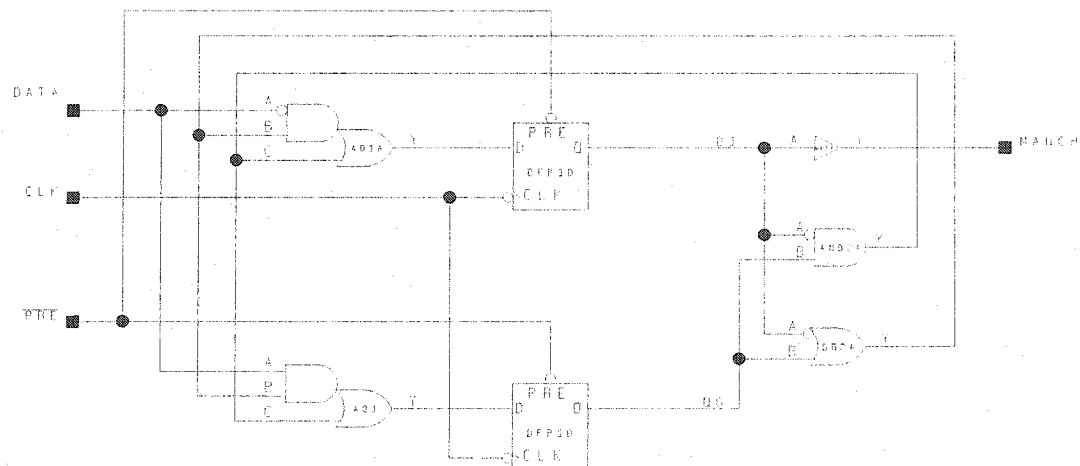
Codeur Manchester
(COD_MANCH)

Figure D.13 : Schéma logique du codeur Manchester du FPGA du module de stabilisation de la porteuse