Réalisation d’un système de mesures d’impédances pour la caractérisation de cellules vivantes en régime dynamique
Nicolas Georges

To cite this version:
Nicolas Georges. Réalisation d’un système de mesures d’impédances pour la caractérisation de cellules vivantes en régime dynamique. Sciences de l’ingénieur [physics]. 2013. hal-01860227

HAL Id: hal-01860227
https://hal.univ-lorraine.fr/hal-01860227
Submitted on 23 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-memoires-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4  
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10  
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm
Université de Lorraine
Faculté des Sciences et Technologies

Master Systèmes Embarqués et énergie
Spécialité « systèmes électroniques embarqués »
Année universitaire 2012/2013

Réalisation d’un système de mesures d’impédances pour la caractérisation de cellules vivantes en régime dynamique

Mémoire présenté par « Georges Nicolas »
Soutenu le 13 septembre 2013

Stage effectué à l’Institut Jean Lamour
Équipe 406 « Mesure Architectures Electroniques »
Faculté des Sciences et Technologies
BP 70239 bâtiment 2ème cycle Entrée 2A 3ème étage
54506 VANDOEUVRE LES NANCY Cedex

Tuteur universitaire : Claudel Julien
Tuteur universitaire : Kourtiche Djilali
Remerciements

Dans un premier temps, je remercie mes tuteurs de stage M. CLAUDEL Julien, Doctorant à l'Université de Lorraine travaillant sur le sujet de thèse sur lequel nous avons réalisé notre projet et M. KOURTICHE Djilali, Professeur à l'Université de Lorraine pour nous avoir introduit clairement le sujet de stage et d'avoir été disponibles pour répondre à mes questions.

Puis dans un second temps, nous remercions l'Université de Lorraine et l'Institut Jean Lamour qui m'ont accueilli et ont su trouver des financements pour l'achat des éléments qui m’ont permis d'avancer sur notre projet.
# Table des matières

**Introduction** .............................................................................................................................................. 4

1. **Présentation du sujet de stage.** ............................................................................................................... 5
   1.1. **Contexte.** ........................................................................................................................................... 5
   1.2. **Travail à réaliser.** ............................................................................................................................... 5
   1.3. **Compétences requises.** ..................................................................................................................... 5

2. **Etude Bibliographique.** .......................................................................................................................... 6
   2.1. **Caractéristiques électriques et diélectriques du milieu.** ................................................................. 6
   2.2. **Techniques de mesure d’impédance.** ................................................................................................. 8

3. **Implémentation du système de mesure.** ............................................................................................ 11
   3.1. **Support utilisé.** ................................................................................................................................ 11
   3.2. **Système de mesure de type détection synchrone.** ......................................................................... 12
   3.3. **Simulation.** ...................................................................................................................................... 13
   3.4. **Réalisation.** ...................................................................................................................................... 14
      3.4.1. **Simulation de la PLL.** ............................................................................................................... 14
      3.4.2. **Simulation et test de la génération des signaux sinusoidaux.** .................................................. 15
      3.4.3. **Test d’envois et de réceptions des signaux sinusoidaux.** ......................................................... 18
      3.4.4. **Test de la multiplication.** ........................................................................................................ 20
      3.4.5. **Implémentation du filtre et conditionnement des données en sortie.** ...................................... 21
      3.4.6. **Implémentation du projet sur la carte.** .................................................................................. 24

4. **Conclusion.** .............................................................................................................................................. 26

5. **Annexes.** .................................................................................................................................................. 28

**Bibliographie** .................................................................................................................................................... 38
Introduction.

Le sang est un fluide vital circulant de façon continue dans les vaisseaux sanguins et le cœur. Ce liquide permet de diffuser le dioxygène ainsi que les éléments nutritifs nécessaires à l’être humain. Il permet aussi de transporter les déchets, tels que le dioxyde de carbone, en vue d’être éliminés par le corps. Il permet également d’amener aux tissus les éléments indispensables au point de vue immunitaire et à diffuser les hormones dans tout l’organisme. En cas d’anomalies chez un individu, il peut être nécessaire de procéder à des analyses de sang afin de déterminer la source du problème. D’autre part, afin d’en savoir plus sur le fonctionnement du corps humain, il peut être utile de connaître les caractéristiques des différentes cellules présentes dans le sang.

Au cours du siècle dernier, un certain nombre de méthodes ont été développées pour déterminer les propriétés électriques et diélectriques des cellules biologiques en suspension dans un liquide tel que le sang. La majorité des techniques développées sont limitées étant donné qu’elles ne donnent qu’une indication globale des caractéristiques électriques et diélectriques des différents composants du milieu mesurés [1]. En effet, chaque élément présent dans le sang a des comportements et des caractéristiques différentes (globules blancs, globules rouges, plaquettes, plasma etc.). C’est pourquoi, les caractéristiques déterminées ne permettent pas une bonne discrimination des caractéristiques de chaque élément.

Depuis peu, avec le développement significatif des micro-technologies, des expériences mettant en évidence les caractéristiques unitaires des différentes cellules présentes dans le sang ont pu être effectuées [1] [2]. Ces expériences nécessitent des outils spécifiques et dédiés.

A travers ce stage, qui s’inscrit dans la continuité d’un projet effectué le semestre dernier, nous allons tenter de réaliser le conditionnement électronique d’un capteur de type micro-fluidique. Il consiste en la génération, le traitement et l’analyse de signaux permettant la caractérisation de cellules à l’aide d’une carte FPGA, permettant la miniaturisation et la diminution des coûts comparé aux appareils standard déjà présents sur le marché.
1. Présentation du sujet de stage.

1.1. Contexte.

Ce stage s’inscrit dans la continuité d’un projet effectué dans le cadre d’un travail de thèse effectué à l’institut Jean Lamour, sur la réalisation d’un capteur micro-fluidique pour la caractérisation de cellules en régime dynamique, par spectroscopie d’impédance. Les informations contenues dans les mesures d’impédance des milieux et tissus biologiques peuvent nous renseigner sur les propriétés physiques et chimiques de ceux-ci. Il est ainsi possible d’utiliser ces mesures comme moyens de diagnostic ou de suivit temporel de l’état d’un tissu (reproduction cellulaire, mort cellulaire…). Ce projet rejoint la dynamique actuelle des Lab-on-Chip (laboratoire sur puce) ; systèmes portatifs permettant des analyses rapides et peu couteuses sur des échantillons de faibles dimensions.

Lors de ce stage, il sera réalisé un outil de mesure d’impédance rapide appliqué à la mesure des caractéristiques de cellules en suspensions (typiquement le cas des cellules du sang). Cet outil sera réalisé grâce à une carte mère FPGA couplée à une carte fille embarquant des convertisseurs analogique/numérique et numérique/analogique haute vitesse ; permettant la génération, l’acquisition et le traitement des signaux de mesure.

1.2. Travail à réaliser.

- Etude bibliographique sur les caractéristiques électriques (conductivité) et diélectrique (permittivité) des milieux biologiques, ainsi que sur les techniques de mesure d’impédance.
- Implémentation d’un système de mesure basé sur le principe de la détection synchrone (Lock-In amplifier).

1.3. Compétences requises.

Afin de mener à bien ce sujet, il est nécessaire d’avoir des connaissances en implémentation FPGA et description VHDL ainsi qu’en programmation C/C++.
2. Etude Bibliographique.

2.1. Caractéristiques électriques et diélectriques du milieu.

Sur le schéma précédent, nous pouvons observer le modèle du circuit équivalent d’une cellule suspendue entre une paire de micro électrode. $R_{DL}$ et $C_{DL}$ représentent la résistance et la capacitance de la double couche électrique\(^1\), celle-ci à une haute impédance en basse fréquence, $R_m$ et $C_m$ représentent la résistance et la capacitance du milieu, $R_{mem}$ et $C_{mem}$ la résistance et la capacitance de la membrane de la cellule, et $R_i$ la résistance du cytoplasme\(^2\). Dans la seconde partie de la figure 2 est représenté le schéma électrique simplifié.

![Figure 3 : Spectre électrique et diélectrique général d'une suspension de cellule.](image)

Sur la figure ci-dessus, Nous pouvons y voir que les effets des différentes parties de ce modèle, se répercutent à des fréquences différentes, et peuvent donc être déterminés par des mesures à des fréquences discrètes.

---

\(^1\)La double couche électrique est un modèle décrivant la variation du potentiel électrique aux abords d'une surface.

\(^2\) Le cytoplasme désigne l’intérieur d’une cellule vivante.
2.2. Techniques de mesure d’impédance.

Le capteur utilisé pour la mesure d’impédance est constitué de deux réservoirs. Un d’entrée pour le dépôt de l’échantillon liquide (sang) et un de sortie pour la réception de ce même échantillon. Ce liquide est déplacé par un système de pompage dans la zone de mesure, qui est composée de microélectrodes en platine dans un canal de section 20um x 20um. La figure ci-dessous schématisé la zone de mesure. La mesure d'impédance doit s'effectuer en temps réel pendant le passage de la cellule au centre de cette zone de mesure. Seules les valeurs d'impédance mesurées lorsque la cellule se situe au centre de la zone de mesure sont utiles. Les figures 4 et 5 schématisent le principe de fonctionnement d’un capteur micro-fluidique.

Figure 4 : Schéma d'un capteur micro-fluidique et d'un système de mesure par spectroscopie d'impédance [1]

Figure 5: Schéma de la zone de mesure [6]
La courbe au-dessus est issue de résultats de simulation. Elle permet d'observer la variation d'impédance à une fréquence donnée lors du passage d'une cellule pour plusieurs géométries d'électrodes. On observe que les mesures "utiles" se situent dans la zone où la variation est maximale. Le système de mesure final doit donc être capable de mesurer des échantillons d'impédance à une fréquence suffisante, afin d'obtenir pour chaque passage de cellule au moins une mesure exploitable.
2.3. Principe de la détection synchrone.

L'amplificateur à détection synchrone (Lock-in Amplifier) permet la mesure de faibles signaux sinusoidaux, même très bruités. Ce mode de mesure est donc parfaitement adapté à la mesure de tissus biologiques, qui nécessitent l'utilisation de tensions et de courants très faibles afin de ne pas altérer ou détruire un échantillon de faibles dimensions ce qui entraîne une forte sensibilité au bruit. Le principe repose sur une démodulation suivant une fréquence porteuse, comme dans le cas d'une transmission de données.

On génère en interne deux signaux de même fréquence mais en quadrature de phase: Vint1 et Vint2. Le premier est utilisé pour générer le signal d'excitation Ve.

Le signal de réponse Vin est ensuite multiplié d'une part par le signal Vint1 et de l'autre par Vint2. Cette action permet d'obtenir deux signaux dont les composantes continues sont respectivement fonction de la partie réelle et de la partie imaginaire du signal mesuré. On obtient ainsi directement la réponse du système après application d'un filtre passe bas à ces signaux.

La figure ci-dessous illustre le principe de fonctionnement d'un système de mesure à détection synchrone, et les calculs (voir "Equations sur les sinus") associés permettent d'éclaircir l'obtention des résultats décrits précédemment.

![Figure 7: Schéma de principe d'un système de mesure par détection synchrone](image-url)
3. Implémentation du système de mesure.

3.1. Support utilisé.

La caractérisation de la cellule doit se faire à l’aide de 2 fréquences au minimum. En effet, une fréquence permet de déterminer la conductivité et la permittivité relative du cytoplasme et l’autre celle de la membrane et de la dimension de la cellule. La fréquence permettant de déterminer l’impédance de la partie externe se situe dans les alentours de 100kHz alors que celle de la partie interne se situe dans les alentours de 10MHz. La carte de développement doit donc être capable de générer et traiter des signaux à ces fréquences. Nous devons donc faire appel à une carte dont la fréquence d’horloge doit pouvoir atteindre au minimum 100MHz. D’après le théorème d’échantillonnage, il faut au minimum une fréquence d’échantillonnage de 20MHz pour un signal de 10MHz. Cependant, il est préférable d’utiliser une fréquence plusieurs fois supérieure.

Les cellules étant des éléments très sensibles, il ne faut pas utiliser de tensions supérieures à 100mV. Ceci dit, pour avoir des signaux interprétables, les tensions ne doivent pas être inférieures à 10mV.

Voici les cartes utilisées qui répondent à nos attentes :

Caractéristiques de la carte mère : DE2–115

- Horloge de 50MHz (possibilité d’augmenter à 100MHz en utilisant une PLL)
- Mémoires incluses : SDRAM (128Mo), SRAM (2Mo), Flash (8Mo), EEPROM (32Kbits)
- Ecran d’affichage LCD (16x2)
- Lecteur de carte SD
- Ports USB Type A & B (dialogue)

Comme nous avions déjà travaillé lors de notre cursus sur des cartes Altera, et que nous disposons de la licence complète pour Quartus, nous avons privilégié les FPGA de chez Altera.
Caractéristiques de la carte fille : AD/DA Data conversion

- Connecteur HSMC
- 2 AD de 14bits à 150MSPS
- 2 DA de 14bits à 250MSPS

Les convertisseurs nous permettent de travailler à des fréquences suffisamment élevées pour pouvoir répondre aux contraintes qu’induit le système de mesure.

3.2. Système de mesure de type détection synchrone.

Figure 8: Schéma du système de mesure
Comme illustré dans la Figure 8, on utilisera un système de mesure deux points pour la détection synchrone, comme les valeurs des impédances sont suffisamment élevées on peut considérer que le courant sera image de la tension aux bornes de la charge. Les convertisseurs se trouvent sur la carte fille, après la charge, il se peut que l'on ajoute un ampli/buffer entre celle-ci et la carte fille pour corriger des problèmes d’adaptation d’impédance et augmenter l’amplitude du signal mesurer si celui-ci est trop faible. L’unité de calcul correspond à-peu-près à celle illustré dans la figure 7, un schéma détaillé sera présenté plus tard. Ensuite les données seront rapatriées via l’unité logique de communication. Les signaux générés par le système de mesures seront une sinusoïde et une autre sinusoïde avec un retard de $\pi/2$.

3.3. Simulation.

Pour mieux cerné les signaux attendu, une simulation sous l’outil de CAO ProtelDXP a été effectué.

Sur le premier graphique, les signaux décrit sont ceux sur lesquels nous devons effectuer des traitements. En bleu, il s'agit du sinus envoyé sur la cellule et en bleu celui déphasé de $\pi/2$. Afin de simplifier la simulation, Vsin1 et Vsin2 ont la même amplitude. Sur le second graphique, Le sinus mesuré est dans un premier temps multiplié par Vsin1 puis par Vsin2. Nous obtenons donc les 2 sinus présents ici en violet et vert. Si on observe la valeur moyenne du signal en vert, on s'aperçoit qu'elle est de 0.5V et qui lorsqu'elle est multipliée par 2 nous donne 1V. D'autre part, si on observe la valeur moyenne du signal en violet, on s'aperçoit qu'elle est de 0V et qui lorsqu'elle est multipliée par 2 donne toujours 0V. En analysant les données ci-dessus, on peut donc dire que la partie réelle est de 1V, la partie imaginaire de 0V et le déphasage de 0.

Figure 9: résultats de la simulation
Afin d'observer la composante continue, il faut effectuer une moyenne des signaux vert et violet en utilisant un filtre passe bas dont la fréquence de coupure est très inférieure à la fréquence des signaux initiaux (Vsin1 et Vsin2).

3.4. Réalisation.

Pour la réalisation de ce projet, j'ai travaillé de manière progressive en décomposant et en testant fonction par fonction chacune des parties.

3.4.1. Simulation de la PLL.

Si l'on souhaite travailler avec des signaux sinusoidaux de fréquences de 10MHz, il faut avoir une fréquence d’horloge de 100Mhz pour pouvoir avoir 10 échantillons par période. La carte Terasic DE2-115 permet de générer jusqu’à 4 PLL. Nous allons utiliser la fonction ALTPLL généré à l’aide de l’outil Megawizard Plug-In Manager d’Altera. A partir de la fréquence d’horloge interne au FPGA de 50MHz, l’outil va nous permettre de générer le signal d’horloge voulu.

Résultats sous ModelSim :

En simulant les signaux de fréquence d’entrée et fréquence de sortie, on s’aperçoit que la PLL fonctionne correctement. En effet, le signal issu de la PLL a bien une période 2 fois plus faible que le signal d’horloge. Les signaux suivants prouvent son bon fonctionnement.

Figure 10 : Simulation de la PLL de 100Mhz sous ModelSim

Les résultats étant satisfaisants, la PLL sera implémenté sur le FPGA.
3.4.2. Simulation et test de la génération des signaux sinusoïdaux.

Pour générer les sinusoïdes, plusieurs solutions étaient possibles. Utiliser une table d’observation (look-up table) pour lire les valeurs et les envoyer vers les convertisseurs étaient la solution la plus simple à mettre en œuvre mais n’est pas évolutive, si nous voulions éditer les valeurs ou la fréquence de lecture des données cela devenait plus compliqué. Nous avons donc généré un sinus à l’aide de la fonction NCO\(^3\) du MegaWizard Plug-In Manager d’Altera.

Dans la rubrique parameters :
AngularResolution => 14
Magnitude Precision => 14
Phase Increment Value => 100MHz
Décrocher : Implement Phase Dithering

Dans la rubrique Implementation :
Target => Cyclone IV E
Cocher: Single Output

Résultats :

En observant le signal généré à l’aide d’un oscilloscope, nous avons remarqué que le signal ressemblait à un sinus avec débordement. Nous avons donc tenté de diminuer le nombre de bits qui était alors de 14 à 13. Le problème a persisté, nous avons constaté que si l’on inversait les 2 « demi-lobes », nous pouvions obtenir un sinus ; Nous avons donc pensé que le bit de signe (bit 13) était inversé. Nous avons donc complété la valeur du bit 13, ce qui s’est traduit par l’obtention d’un sinus.

\(^3\) Numerically Controlled Oscillator (oscillateur controlé numériquement)
Nous avons ensuite tenté de générer un sinus à la fréquence de 1MHz puis à 10MHz. A 1MHz, nous avons obtenu un sinus à l’oscilloscope sans difficultés alors que pour la fréquence de 10MHz, nous avons éprouvé des difficultés étant donné que le NCO créait les échantillons suivants en fonctions des précédents (récursivité). Par la suite, il faudra trouver une méthode de génération de sinus qui elle sera non récursive afin d’éviter ce genre de soucis. C’est pourquoi, il suffit qu’un échantillon soit mal déterminé pour que tout le sinus soit altéré. Nous avons relevé les signaux suivants à l’aide de l’oscilloscope :

![Sinusoïde à 10 MHz mal capturée](image13.png)  ![Sinusoïde à 10MHz bien capturée](image14.png)

Suite à un changement de matériel, avec le nouvel oscilloscope à large bande de fréquence, les irrégularités n’étaient plus visibles. Cela venait donc de la bande passante limitée de l’oscilloscope
Dans un second temps, nous avons essayé de générer un second sinus qui lui était déphasé de Pi/2 par rapport au premier.
Pour se faire, il a fallu choisir la fonction « NCO » dans DSP => Signal Generation => NCO. En gardant les paramètres initiaux, il a fallu modifier :

Dans la rubrique parameters :
AngularResolution => 14
Magnitude Precision => 14
Phase Increment Value =>100MHz
Décrocher : Implement Phase Dithering

Dans la rubrique Implementation :
Target => Cyclone IV E
Cocher: Phase Modulation Input
ModulatorPrecision => 32
Modulator Pipeline Level => 1
Cocher: Single Output
Résultats :

De la même manière que précédemment, il a fallu inverser le bit de signe afin d’obtenir un sinus. Par défaut, la valeur maximum de « phase_mod » correspond à un retard de 2πi. Pour obtenir un retard de π/2, il a fallu diviser la valeur de « phase_mod » par 4 en mettant à 1 le 31ème bit et tous les autres à 0 ; Nous avons obtenu un sinus en quadrature.

![Figure 15: Sinus et sinus retardé de π/2](image)
3.4.3. **Test d’envoi et de réceptions des signaux sinusoïdaux.**

Une fois la génération des signaux effectuée nous avons testé l’envoi et la réception du sinus via la carte AD/DA Data conversion.
En sortie des CAN et en entrée des CNA se trouve des transformateurs de chez Mini-Circuit.

![Surface Mount RF Transformer](image)

**Figure 16: Partie des spec. technique des transformateurs**

On peut constater que le transformateur permet une adaptation d’impédance entre entrée et sortie en appliquant à celles-ci une impédance de 50 Ω. Autre spécification il agit comme un filtre passe haut en supprimant les valeurs continues.
A la réception, pour avoir des valeurs correspondant au signal réel et pouvoir effectuer les calculs il faut convertir le signal reçu en nombre signé en complément à deux. Pour cela il faut :
Inverser les valeurs des bits reçus et ajouter la valeur ‘1’.
De même, il faut répéter la même opération pour les signaux internes qui étaient des nombres non signés.
Figure 17: Aperçu des résultats de réception et de multiplication des signaux via SignalTap Logic Analyzer

Dans la figure ci-dessus, on peut observer 6 signaux différents, ils correspondent aux signaux obtenus lorsque le CNA est relié directement au CAN. Le premier est la sinusoïde envoyée (en nombre non signé) via le CNA ; $u_{\cos1}$ est le signal non signé correspondant à la sinusoïde retardée de $\pi/2$ ; ADA est le signal reçu par le CAN, observé à sa sortie, il est systématiquement inversé : le signal dans son cheminement traverse deux transfo que l’on peut considérer comme élément purement inductif, chacun retarde donc le signal de $\pi/2$.

Au maximum, le signal ADA à une valeur de 6152 alors qu’il est censé être sans perte (soit 8192 ou $2^{13}$), cela est dû aux transformateurs, ceux-ci induisent une baisse de niveau on peut évaluer la perte à hauteur de 2.48 dB :

<table>
<thead>
<tr>
<th>Transformer Electrical Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ROUTIO</strong></td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

* Insertion Loss is specified with input at pin 4 and output at pin 1 with pins 6 & 3 grounded and pins 2 & 5 open.

---

4 Outil permettant d’observer les signaux interne au FPGA
Études d’effets de la fréquence du signal

Figure 18: Pertes dans les transformateurs suivant la fréquence du signal

Ces signaux sont de fréquence de 500KHz et contiennent 10 échantillons par période. Nous aurions pu augmenter le nombre d’échantillons mais la nature du lock-in amplifier nous permet de nous contenter de ça, ce qui nous permet un gain de place sur le FPGA ainsi que de vitesse de traitement des données.

Ces observations permettent de constater que la transmission des signaux s’effectue correctement. Les signaux suivant correspondent à la multiplication du signal reçu par le CAN et du sinus généré en interne.

3.4.4. Test de la multiplication.

Cette première multiplication va nous permettre de réaliser le produit entre le signal reçu par le CAN et les signaux générés en interne (voir figure 7).

Cette fonction est réalisée à l’aide de la fonction LMP_MULT de l’outil MegaWizard Plug-In Manager d’Altera. Elle permet une certaine souplesse pour le choix du nombre de bits en entrée et en sortie ainsi que le type de multiplication (signé, non signé, etc…).

Figure 19: Signaux servant à la multiplication

En effectuant le calcul analytiquement nous observons sur le même résultat :

\[ 6152 \times 8141 = 50083432 \]

Le produit entre les deux signaux est bien effectué.
3.4.5. **Implémentation du filtre et conditionnement des données en sortie.**

Le schéma en figure 7 n’étant pas complet pour en déduire la valeur de l’impédance mesuré, en voici un schématisant la suite du calcul :

Les équations en figure 25 montrent que l’équation après filtrage est la suivante : 

\[ \frac{A^2 G}{2} \times \cos \varphi \]

C’est pour cela que l’on multiplie par 2 le signal et par la valeur maximal du signal d’entrée, c’est-à-dire A. Au final nous obtenons la valeur \( A G \cos \varphi \) correspondant à la partie continue du signal.

Pour réaliser le filtre, nous utiliserons la fonction FIR Compiler du MegaWizard d’Altera. Cette fonction est pratique pour pouvoir configurer facilement la fréquence de coupure et le nombre de coefficient.

Un filtre FIR est un filtre à réponse impulsionnel fini, sont principe de fonctionnement est le suivant :

![Diagramme du filtre FIR](image)

*Figure 20: Structure quelconque d’un filtre FIR*
Le filtre que nous avons réalisé pour filtrer le signal à 500KHz est paramétré de la manière suivante :

- C’est un filtre passe bas
- Il possède 100 coefficients
- Sa fréquence de coupure est de 20KHz
- Sa fréquence d’échantillonnage est de 5 MHz

On peut observer que le signal est atténué de 35dB à 10% de la fréquence d’échantillonnage en lissant la courbe, ce qui dans notre cas est suffisant pour récupérer la valeur du signal continu. Mais ceci dit, il faudra tenir compte d’une légère atténuation du signal, de l’ordre de 2dB.

En pleine résolution, le signal de sortie du filtre est codé sur 62 bits ce qui donne une valeur de l’ordre de $2 \times 10^{18}$ ce qui n’est pas vraiment exploitable dans l’état.

Suite à une recherche sur le forum d’aide d’Altera, j’ai trouvé une formule permettant de réduire le nombre de bits en tronquant les bits de poids faible non nécessaire :

$$\text{Nb bits} = \log_2 \left( (2^{28} - 1) \times \sum \text{coefficients} \right) = 27.53$$

Ce qui nous donne un total de 28 bits, ce qui revient à tronquer 34 LSB ($2^{28} - 1$ correspond à la valeur maximal en entrée du filtre).
Ci-dessous se trouve les résultats du filtrage et des opérations suivantes.

Nous pouvons constater que la valeur en sortie du filtre est continue sur plusieurs périodes. La suite des calculs est réalisée avec des fonctions simples (opérations mathématiques) qui ne sont pas nécessaire de présenter plus tard. Les résultats sont facilement démontrables de façon analytique.

Le signal finale correspondant à $A \cos \varphi$ est le signal nommé « div : div_r_20k|quotient ». Ici, il est calculé sans prendre en compte la baisse de niveau dans le continu du au filtre.
3.4.6. Implémentation du projet sur la carte.

L’étape suivante est de mettre toutes ces fonctions sur la carte FPGA et la configurer de façon à ce qu’elle puisse générer des signaux de 500KHz et 5 MHz. Et qu’elle puisse effectuer les mesures et calculs pour les parties réels et imaginaires de ses fréquences. Ce qui implique de mettre 4 multiplicateurs de 2 fois 14 bits en entrées et 28 bits en sorties, 2 filtres de fréquences de coupure de 20KHz, 2 filtres de fréquences de coupure 200KHz (il suffit de multiplier par 10 la fréquence d’échantillonnage et la fréquence de coupure), 4 multiplicateurs par 2, et 4 diviseurs. Divers switches et boutons ont été configurés pour gérer les reset des fonctions ainsi que le choix de la plage de fréquences et l’inversion de certains signaux en cas de besoins (voir annexes). Les choix de fréquences ci-dessus nous permettent de ne pas utiliser de PLL. Il a fallu aussi effectuer une table de correspondance entre la carte d’acquisition et la carte FPGA, celle-ci n’étant pas fournie par le constructeur. Il a fallu analyser les différentes documentations constructeur et en faire le lien (voir annexes).

Une fois tout cela effectuée, nous pouvons compiler le projet FPGA et voici le résultat de la compilation :

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Quartus II 5+Bit Version</td>
<td>13.0.0 Build 156 04/24/2013 SJ Full Version</td>
</tr>
<tr>
<td>Revision Name</td>
<td>lockin_amp</td>
</tr>
<tr>
<td>Top-level Entity Name</td>
<td>lockin_amp</td>
</tr>
<tr>
<td>Family</td>
<td>Cyclone IV E</td>
</tr>
<tr>
<td>Device</td>
<td>EP4CE115F29C8</td>
</tr>
<tr>
<td>Timing Models</td>
<td>Fnet</td>
</tr>
<tr>
<td>Total logic elements</td>
<td>10,171 / 114,480 ( 9 %)</td>
</tr>
<tr>
<td>Total combinational functions</td>
<td>6,414 / 114,480 ( 6 %)</td>
</tr>
<tr>
<td>Dedicated logic registers</td>
<td>9,050 / 114,480 ( 8 %)</td>
</tr>
<tr>
<td>Total registers</td>
<td>9050</td>
</tr>
<tr>
<td>Total pins</td>
<td>76 / 529 ( 14 %)</td>
</tr>
<tr>
<td>Total virtual pins</td>
<td>0</td>
</tr>
<tr>
<td>Total memory bits</td>
<td>321,343 / 3,981,312 ( 8 %)</td>
</tr>
<tr>
<td>Embedded Multiplier 9-bit elements</td>
<td>22 / 532 ( 4 %)</td>
</tr>
<tr>
<td>Total PLLs</td>
<td>0 / 4 ( 0 %)</td>
</tr>
</tbody>
</table>

Figure 23: Résultats de la compilation FPGA

Les différents choix effectués nous permettent d’utiliser qu’une petite partie des ressources du FPGA.
3.4.7. Mesures d’impédances.

Maintenant que la carte FPGA est configurée pour réaliser les mesures, nous allons effectuer des tests avec des résistances de 12kΩ et 22kΩ, symbolisé par la charge Z sur le schéma ci-dessus. Ici on aura configuré la résistance d’entrée de l’amplificateur à 50Ω et son gain G à 40dB (x100). Il nous donnera en sortie une image du courant traversant Z.

En négligeant les impédances des transformateurs et de la résistance d’entrée de l’amplificateur par rapport à l’impédance de Z, on peut estimer analytiquement la valeur de Z à partir de $V_e$ et $V_{mes}$.

\[ i_e = \frac{v_e}{Z + 50} \approx \frac{v_e}{Z} \]

\[ v_{mes} = i_e \times 50 \times G \]

\[ Z = \frac{v_e}{i_e} = \frac{v_e}{v_{mes}} \times 50 \times G \]

Maintenant nous allons tester le bon fonctionnement du projet FPGA. $V_{mes}$ correspondra au signal ADA vu précédemment et $V_e$ sera le signal en sortie du CNA. Pour les résultats voir annexes les figures 28 et 29.

Ici, pour les calculs nous appellerons $1/K$ le gain apporté par le filtre FIR comme vu précédemment et $V_{out}$ le signal de sortie du filtre une fois traité.

Pour retrouver l’impédance par rapport à la formule précédente on prendra $V_{mes} = V_{out} \times K$.

A partir de cela nous allons déterminer $K$.

$V_{out}$ et $V_{out\_im}$ sont les valeurs des parties imaginaires et réelles

On a donc $V_{out} = \sqrt{V_{out\_im}^2 + V_{out\_re}^2}$

Vu les résultats on a pour une résistance de 22KΩ et un gain de 100 on a $V_{out} = 1211$ (valeur interne au FPGA en nombre entier signé)

A partir de cette équation : $|Z| = \frac{v_e}{V_{out\times K}} \times 50 \times G$ on peut déterminer $K$ :

\[ K = \frac{v_e \times 5000}{Z \times V_{out}} = 1.53 \]

Vu les résultats on a pour une résistance de 12KΩ et un gain de 100 on a $V_{out} = 2192$ (valeur en INT)

A partir de cette équation : $|Z| = \frac{v_e}{V_{out\times K}} \times 50 \times G$ on peut déterminer $K$ :
$K = \frac{v_e \times 5000}{Z \times v_{out}} = 1.55$

En visualisant les deux résultats précédents on peut constater que K est constant donc on peut se permettre de dire que nous venons de caractériser le facteur K du filtre FIR ce qui donne une perte de -3.5 dB ($20\log_{10} (1/K)$).

Nous aurions pu essayer des mesures sur une impédance capacitive.

Pour la suite des calculs, pour caractériser l’impédance il faudrait appliquer un facteur de $50 \times G/K$.

4. Conclusion.

A partir d’une carte FPGA et d’une carte d’acquisition générique, nous avons pu réaliser un système spécifique de mesures d’impédances pour la caractérisation de cellules vivantes en régime dynamique en utilisant un outil de détection synchrone. Les caractéristiques de la chaîne de mesure ont pu être modélisées et validées. Il nous permet effectivement de mesurer de faibles signaux bruités sur la plage de fréquence demandé. Ce qui nous permet de dire que ce dispositif est apte à caractériser les cellules.

Quelques imprévus et un manque de temps nous a empêché d’effectuer toutes les mesures et les fonctions souhaité. Effectivement nous aurions pu effectuer l’implémentation de la dernière partie des calculs ainsi que créer une interface PC-FPGA pour rapatrier les données.

Dans un avenir proche, ce travail pourrait servir de base pour la création d’un système compact et autonome de mesure d’impédance avec l’association d’un capteur micro-fluidique.
Figure 1 : Principaux composants du sang [5] ................................................................. 6
Figure 2 : Modèle électrique d'une cellule et son schéma simplifié [1] ............................... 6
Figure 3 : Spectre électrique et diélectrique général d'une suspension de cellule .............. 7
Figure 4 : Schéma d'un capteur micro-fluidique et d'un système de mesure par spectroscopie d'impédance [1] .................................................................................................. 8
Figure 5 : Schéma de la zone de mesure [6] .................................................................................. 8
Figure 6 : Variation de l'impédance lors du passage d'une cellule dans la zone de mesure .... 9
Figure 7 : Schéma de principe d'un système de mesure par détection synchrone .............. 10
Figure 8 : Schéma du système de mesure ................................................................................. 12
Figure 9 : Résultats de la simulation .......................................................................................... 13
Figure 10 : Simulation de la PLL de 100Mhz sous ModelSim ................................................. 14
Figure 11 : Première tentative de génération de sinusoides .................................................... 15
Figure 12 : Deuxième tentative de génération de sinusoides .................................................... 15
Figure 13 : Sinusoides à 10 MHz mal capturée ........................................................................ 16
Figure 14 : Sinusoides à 10 MHz bien capturée ....................................................................... 16
Figure 15 : Sinus et sinus retardé de PI/2 ................................................................................ 17
Figure 16 : Pertes dans les transformateurs suivant la fréquence du signal .......................... 20
Figure 17 : Aperçu des résultats de réception et de multiplication des signaux via SignalTap Logic Analyzer ........................................................................................................... 19
Figure 18 : Pertes dans les transformateurs suivant la fréquence du signal .......................... 20
Figure 19 : Signaux servant à la multiplication ........................................................................ 20
Figure 20 : Structure quelconque d'un filtre FIR .................................................................... 21
Figure 21 : Réponse fréquentielle du filtre FIR ....................................................................... 22
Figure 22 : Résultats du filtrage et des opérations permettant de récupérer la valeur correspondant à la partie réelle de l'impédance mesurée ......................................................... 23
Figure 23 : Résultats de la compilation FPGA ........................................................................ 24
Figure 24 : Schéma de la charge ............................................................................................... 25
Figure 25 : Equations sur les sinus .......................................................................................... 28
Figure 26 : Pin planer et le tableau associé ............................................................................ 29
Figure 27 : tableau de correspondance entre Carte mère et carte fille ................................ 30
Figure 28 : Résultats des mesures pour une résistance de 12kOhms .................................. 31
Figure 29 : Résultats des mesures pour une résistance de 22kOhms .................................... 32
5. Annexes.

Equations des deux signaux internes :

\[ V_{\text{int}_1} = A \sin(\omega t) \]
\[ V_{\text{int}_2} = A \sin(\omega t + \frac{\pi}{2}) \]

Equations du signal mesuré :

\[ V_{\text{in}} = A \! G \sin(\omega t + \phi) + b(t) \]

Avec G le gain du système (et des éventuels amplificateurs en entrées et sorties), \( \phi \) le déphasage par rapport au signal interne \( V_{\text{int}_2} \) et \( b(t) \) les bruits et perturbations.

Calcul réalisé en multipliant \( V_{\text{int}_1} \) et \( V_{\text{in}} \) :

\[ A \! \sin(\omega t) \cdot [A \! G \sin(\omega t + \phi) + b(t)] \]
\[ \frac{A^2 G}{2} \left[ \cos(\omega t - \phi) - \cos(2 \omega t + \phi) \right] + A \sin(\omega t) b(t) \]
\[ \frac{A^2 G}{2} \cos(\phi) - \frac{A^2 G}{2} \cos(2 \omega t + \phi) + A \sin(\omega t) b(t) \]

On obtient un signal avec une composante continue \( \frac{A^2 G}{2} \cos(\phi) \).

L'amplitude de la partie réelle du signal est déterminée après filtrage passe bas et division par \( A/2 \) :

\[ \text{Re}(V_{\text{in}}) = A \! G \cos(\phi) \].

Calcul réalisé en multipliant \( V_{\text{int}_1} \) et \( V_{\text{in}} \) :

\[ A \sin(\omega t + \frac{\pi}{2}) \cdot [A \! B \sin(\omega t + \phi) + b(t)] \]
\[ \frac{A^2 B}{2} \left[ \cos(\omega t - \phi + \frac{\pi}{2}) - \cos(2 \omega t + \phi + \frac{\pi}{2}) \right] + A \sin(\omega t + \frac{\pi}{2}) b(t) \]
\[ \frac{A^2 B}{2} \cos(\phi + \frac{\pi}{2}) - \frac{A^2 B}{2} \cos(2 \omega t + \phi + \frac{\pi}{2}) + A \sin(\omega t + \frac{\pi}{2}) b(t) \]
\[ \frac{A^2 B}{2} \sin(\phi) + \frac{A^2 B}{2} \sin(2 \omega t + \phi) + A \cos(\omega t) b(t) \]

On obtient un signal avec une composante continue \( -\frac{A^2 G}{2} \sin(\phi) \).

L'amplitude de la partie imaginaire du signal est déterminée après filtrage passe bas et division par \( -A/2 \) :

\[ \text{Im}(V_{\text{in}}) = A \! G \cos(\phi) \].

Figure 25: Equations sur les sinus
Figure 26: Planer et le tableau associé
Figure 27: table de correspondance entre Carte mère et carte fille
Figure 28: Résultat des mesures pour une résistance de 12kOhms
Figure 29: Résultat des mesures pour une résistance de 22kOhms
Description VHDL de la fonction principale :

library IEEE;
use IEEE.std_logic_1164.all;
use work.all;

Entity lockin_amp is
  Port( clk : in std_logic; -- horloge du FPGA
        ADA, ADB : in std_logic_vector (13 downto 0); -- Sorties des CAN de la carte fille
        ADA_DCO, ADB_DCO : in std_LOGIC; -- CAN Data Clock Output
        FPGA_CLK_A_P, FPGA_CLK_A_N, FPGA_CLK_B_P, FPGA_CLK_B_N : inout std_LOGIC; -- Non-inverting & inverting
        XT_IN_P, XT_IN_N : in std_LOGIC; -- Non-inverting & inverting Differential clock input
        AD_SCLK, AD_SDIO : inout std_LOGIC; -- Output Enable (Active Low)
        ADA_SPI_CS, ADB_SPI_CS : out std_LOGIC; -- Serial Port Interface Chip Select (Active Low)
        DA, DB : out std_logic_vector (13 downto 0); -- Entrees des CNA
        KEY0: in std_LOGIC; -- reset FIR
        SW0 : in std_LOGIC; -- choix de la fréquence
        sw1 : in std_LOGIC; -- bit signé ou non
        sw2 : in std_logic; -- inversion ADA
        sw3 : in std_logic);-- inversion ADA
End Entity;

Architecture arch of lockin_amp is

component NCO1 IS -- Sin_500KHz
  PORT ( phi_inc_i : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
          clk : IN STD_LOGIC;
          reset_n : IN STD_LOGIC;
          clken : IN STD_LOGIC;
          fsin_o : OUT STD_LOGIC_VECTOR (13 DOWNTO 0);
          fcos_o : OUT STD_LOGIC_VECTOR (13 DOWNTO 0);
          out_valid : OUT STD_LOGIC);
END component;

component NCO2 IS -- sin_5MHz
  PORT ( phi_inc_i : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
          clk : IN STD_LOGIC;
          reset_n : IN STD_LOGIC;
          clken : IN STD_LOGIC;
          fsin_o : OUT STD_LOGIC_VECTOR (13 DOWNTO 0);
          fcos_o : OUT STD_LOGIC_VECTOR (13 DOWNTO 0);
          out_valid : OUT STD_LOGIC);
END component;

Component Mult -- déclaration du multiplicateur
  PORT ( dataa  : IN STD_LOGIC_VECTOR (13 DOWNTO 0);
          datab  : IN STD_LOGIC_VECTOR (13 DOWNTO 0);
          result  : OUT STD_LOGIC_VECTOR (27 DOWNTO 0));
END Component;

Component fir_pb IS -- fir_20KHz
  PORT ( clk : IN STD_LOGIC;
          reset_n : IN STD_LOGIC;
          ast_sink_data : IN STD_LOGIC_VECTOR (27 DOWNTO 0);
          ast_sink_valid : IN STD_LOGIC;
          ast_source_ready : IN STD_LOGIC;
          ast_sink_error : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
          ast_source_data : OUT STD_LOGIC_VECTOR (27 DOWNTO 0);
          ast_sink_ready : OUT STD_LOGIC);
END Component;
component fir_pb2 IS -- fir_200KHz
PORT (
    clk : IN STD_LOGIC;
    reset_n : IN STD_LOGIC;
    ast_sink_data : IN STD_LOGIC_VECTOR (27 DOWNTO 0);
    ast_sink_valid : IN STD_LOGIC;
    ast_source_ready : IN STD_LOGIC;
    ast_sink_error : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
    ast_source_data : OUT STD_LOGIC_VECTOR (27 DOWNTO 0);
    ast_sink_ready : OUT STD_LOGIC;
    ast_source_valid : OUT STD_LOGIC;
    ast_source_error : OUT STD_LOGIC_VECTOR (1 DOWNTO 0)
);
END Component;

component div IS
PORT
( 
    denom  : IN STD_LOGIC_VECTOR (13 DOWNTO 0);
    numer  : IN STD_LOGIC_VECTOR (30 DOWNTO 0);
    quotient  : OUT STD_LOGIC_VECTOR (30 DOWNTO 0);
    remain  : OUT STD_LOGIC_VECTOR (13 DOWNTO 0)
);
END component;

component add IS -- addition pour complément à 2
PORT
( 
    datab  : IN STD_LOGIC_VECTOR (13 DOWNTO 0);
    result  : OUT STD_LOGIC_VECTOR (13 DOWNTO 0)
);
END component;

component mult2 IS -- mult par 2
PORT
( 
    dataa  : IN STD_LOGIC_VECTOR (27 DOWNTO 0);
    result  : OUT STD_LOGIC_VECTOR (30 DOWNTO 0)
);
END component;

signal reset_1 : std_logic := '0'; -- reset pll
signal reset_2 : std_logic ; -- reset NCO
signal clken_1 : std_logic ; -- clock enable sin_500KHz
signal clken_2 : std_logic ; -- clock enable sin_5MHz
signal phi_inc_1 : STD_LOGIC_VECTOR (31 DOWNTO 0); -- phase increment sin_500KHz
signal phi_inc_2 : STD_LOGIC_VECTOR (31 DOWNTO 0); -- phase increment sin_5MHz
signal u_cos, u_sin : std_logic_vector (13 downto 0); -- signaux envoyés vers CNA
signal u.cos1, u.sin1 : std_logic_vector (13 downto 0); -- signaux envoyés vers CNA
signal u.sin.add : std_logic_vector (12 downto 0);
signal sortie_ada : std_LOGIC_VECTOR (13 downto 0); 
signal ada_o : std_LOGIC_VECTOR (13 DOWNTO 0);  
signal sin_nco1 : STD_LOGIC_VECTOR (13 DOWNTO 0);  -- sortie NCO1 
signal cos_nco1 : STD_LOGIC_VECTOR (13 DOWNTO 0); -- sortie cos NCO1 
signal sin_nco2 : STD_LOGIC_VECTOR (13 DOWNTO 0); -- sortie sin NCO2 
signal cos_nco2 : STD_LOGIC_VECTOR (13 DOWNTO 0); -- sortie cos NCO2 
signal amp : STD_LOGIC_VECTOR (13 DOWNTO 0); -- amplitude sinus 
signal tempo1 : STD_LOGIC_VECTOR (13 DOWNTO 0); -- temporisation pour décalage 
signal tempo2 : STD_LOGIC_VECTOR (13 DOWNTO 0); 
signal tempo3 : STD_LOGIC_VECTOR (13 DOWNTO 0); 
signal tempo4 : STD_LOGIC_VECTOR (13 DOWNTO 0); 
signal tempo5 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo6 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo7 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo8 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo9 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo10 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo11 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo12 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo13 : STD_LOGIC_VECTOR (13 DOWNTO 0);
signal tempo14 : STD_LOGIC_VECTOR (13 DOWNTO 0);

signal out_mult_r, out_mult_i : STD_LOGIC_VECTOR (27 DOWNTO 0); -- sorties multiplieurs
signal out_fir1_r, out_fir1_i, out_fir2_r, out_fir2_i : STD_LOGIC_VECTOR (27 DOWNTO 0);
signal out_div_200_r, out_div_20_r, out_div_200_i, out_div_20_i : STD_LOGIC_VECTOR (30 DOWNTO 0);
signal out_ech_200_r, out_ech_20_r, out_ech_200_i, out_ech_20_i : STD_LOGIC_VECTOR (30 DOWNTO 0);

for all : NCO1 use entity work.NCO1(SYN);
for all : NCO2 use entity work.NCO2(SYN);
for all : Mult use entity work.Mult(Syn);
for all : fir_pb use entity work.fir_pb(Syn);
for all : fir_pb2 use entity work.fir_pb2(Syn);
for all : div use entity work.div(Syn);
for all : add use entity work.add(Syn);
for all : Mult2 use entity work.Mult2(Syn);

begin

phi_inc_1 <= "00011001100110011001100110011010"; -- définit la fréquence du sinus
phi_inc_2 <= "00011001100110011001100110011010"; -- définit la fréquence du sinus

amp <= "01111111111111111111111111111111";

sin_500KHz : NCO1 port map (phi_inc_1, sig_clk, reset_2, clken_1, cos_nco1, sin_nco1);

sin_5MHz : NCO2 port map (phi_inc_2, sig_clk, reset_2, clken_2, cos_nco2, sin_nco2);

multiplication_r : Mult port map (sortie_ada, not(u_sin(13))& u_sin(12 downto 0), out_mult_r);

filtre_200k_r : fir_pb port map (sig_clk, reset_2, out_mult_r, '1', '1', '00', out_fir1_r);

filtre_200k_i : fir_pb port map (sig_clk, reset_2, out_mult_i, '1', '1', '00', out_fir1_i);

filtre_20k_r : fir_pb port map (sig_clk, reset_2, out_mult_r, '1', '1', '00', out_fir2_r);

filtre_20k_i : fir_pb port map (sig_clk, reset_2, out_mult_i, '1', '1', '00', out_fir2_i);

div_r_200k : div port map (amp, out_mult2_200_r, out_div_200_r);

div_i_200k : div port map (amp, out_mult2_200_i, out_div_200_i);

clken_1 <= '1';
clken_2 <= '1';
DA <= u_sin1;
DB <= u_sin1;

FPGA_CLK_A_P <= clk;
FPGA_CLK_A_N <= not clk;
FPGA_CLK_B_P <= clk;
FPGA_CLK_B_N <= not clk;

sw_ada : process (sw3, ADA) -- procédure servant à inverser le signal ADA
begin
if (sw3='1') then
   ada_o <= ADA;
else
   ada_o <= not(ADA);
end if;
end process;

sw_freq : process (clk) -- procédure servant à diviser par 10 l'horloge
begin
if rising_edge(clk) then
   if (counter = 5) then
      sub_clk <= NOT(sub_clk);
      counter <= 0;
   else
      counter <= counter + 1;
   end if;
end if;
end process;

sw_freq2 : process (clk) -- procédure servant à créer une horloge pour un échantillonnage à 10KHz
begin
if rising_edge(clk) then
   if (counter_2 = 2500) then
      clk_10KHz <= NOT(clk_10KHz);
      counter_2 <= 0;
   else
      counter_2 <= counter_2 + 1;
   end if;
end if;
end process;

freq : process(SW0, sin_nco1, cos_nco1 , sin_nco2, cos_nco2, sub_clk ) -- procédure pour choix des fréquences
begin
if (SW0 = '1') then
   u_sin1 <= not(sin_nco1(13)) & sin_nco1(12 downto 0);
   u_cos1 <= not(cos_nco1(13)) & cos_nco1(12 downto 0);
   sig_clk <= sub_clk;
else
   u_sin1 <= not(sin_nco2(13)) & sin_nco2(12 downto 0);
   u_cos1 <= not(cos_nco2(13)) & cos_nco2(12 downto 0);
   sig_clk <= clk;
end if;
end process;

reset_nco : process (KEY0) -- procédure pour les reset
begin
if (key0 = '1') then
   reset_2 <= '1';
else
   reset_2 <= '0';
end if;
end process;

temp1 : process(clk) -- procédure pour la temporisation des signaux
begin
if rising_edge(clk) then
   tempo1 <= u_sin1;
   tempo2 <= tempo1;
   tempo3 <= tempo2;
   tempo4 <= tempo3;
   tempo5 <= tempo4;
   tempo6 <= tempo5;
end if;
end process;
tempo7 <= tempo6;
  tempo8 <= u_cos1;
  tempo9 <= tempo8;
  tempo10 <= tempo9;
  tempo11 <= tempo10;
  tempo12 <= tempo11;
  tempo13 <= tempo12;
  tempo14 <= tempo13;
end if;
end process;
u_cos <= tempo14;
u_sin <= tempo7;

ech : process (clk_10KHz, out_div_200_r, out_div_200_i, out_div_20_r, out_div_20_i)
begin
  if rising_edge(clk_10KHz) then -- procedure pour échantillonner les signaux de sortie à 10KHz
    out_ech_200_r <= out_div_200_r;
    out_ech_200_i <= out_div_200_i;
    out_ech_20_r <= out_div_20_r;
    out_ech_20_i <= out_div_20_i;
  end if;
end process;
END arch;
Bibliographie


RESUME :

A travers ce stage, nous allons tenter de réaliser le conditionnement électronique d’un capteur de type micro-fluidique. Il consiste en la génération, le traitement et l’analyse de signaux permettant la caractérisation de cellules à l’aide d’une carte FPGA, permettant la miniaturisation et la diminution des coûts comparé aux appareils standard déjà présents sur le marché.

MOTS-CLES :

FPGA, Spectroscopie d’impédance, amplificateur à détection synchrone

ABSTRACT:

Through this internship, we will try to realize the electrical conditioning of a microfluidic sensor. This project consist in the generation, traitement and analysis of signals that will allow us to characterise living-cells using a FPGA.

KEYWORDS:

FPGA, dielectric spectroscopy, lock-in amplifier