

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد -

تلمس -

Université Aboubakr Belkaïd - Tlemcen -

Faculté de TECHNOLOGIE



## THESE

Présentée pour l'obtention du **grade** de **DOCTEUR EN SCIENCES**

**En** : Génie Biomédical

**Spécialité** : Electronique Biomédical

**Par** : Mme BABA HAMED Amel née BENSEMAIN

### Sujet

**Vers Un Modèle De Classification Neuronale Des Données Médicales A Base De La Technologie FPGA**

Soutenue publiquement, en 2017, devant le jury composé de :

Mr BESSAID Abdelhafid	Professeur	Univ. Tlemcen	Président
Mr CHIKH Mohammed Amine	Professeur	Univ. Tlemcen	Directeur de thèse
Mr RAHMOUNE Abdelatif	Professeur	Ecole Préparatoire d'Informatique. Sidi Bel Abbès	Examineur 1
Mr KERAI Salim	MCA	Univ. Tlemcen	Examineur 2
Mr MERAD Lotfi	MCA	Ecole Supérieur en Sciences Appliquées. Tlemcen	Examineur 3
Mr SOUIER Mahdi	MCA	Ecole Supérieur d'Economie Mangement. Tlemcen	Examineur 4
Mr BECHAR Hassane	MAA	Univ. Tlemcen	Invité 1
Mme BENSEGHIR Sihem Amel	MCB	Univ. Tlemcen	Invitée 2



## *Dédicaces*

*Je dédie cette thèse*

*À la mémoire de mon très cher père ALLAH yerahmek.*

*À ma très chère mère.*

*À mon très cher époux Amine, qui a été toujours là à mes côtés.*

*À mes très chers et adorables enfants*

*Djihane, Imad-Eddine et Alae Nouha*

*À mes très chers frères, ainsi que leurs petites familles.*

*À ma très chère et unique sœur, et sa petite famille.*

*À tous mes amis, en particulier : Nadjet, Nawel, Hanane, Amel et Ilhem.*

*À tous mes collègues de la Faculté de Technologie  
en particulier les laborantins du département de Génie Biomédical.*



*Amel*

## **Remerciements**

*Je remercie **ALLAH** tous puissants, qui m'a donné la force, la patience, le courage et qui m'a permis de concilier entre ma vie familiale, professionnelle et mes études.*

*Je témoigne ma gratitude à mon encadreur Monsieur **M.A CHIKH**, professeur à l'université Abou Bakr Belkaid de Tlemcen, pour son encadrement, son soutien, ses précieux conseils, ses encouragements et ses grandes qualités humaines.*

*Mes très vifs remerciements accompagnés de toute ma gratitude vont à mon co-encadreur Monsieur **H.BECHAR**, maître-assistant à l'université Abou Bakr Belkaid de Tlemcen, pour son soutien, pour ses conseils efficaces et ses grandes qualités humaines. Je lui serai toujours reconnaissante pour l'effort qu'il a investi dans cette thèse.*

*Je remercie Monsieur **A.BESSAID**, professeur à l'université Abou Bakr Belkaid de Tlemcen, pour l'honneur qu'il m'a fait en acceptant de présider le jury de cette thèse.*

*Je remercie vivement Monsieur **A. RAHMOUN**, professeur à l'école Supérieure d'Informatique de Sidi Bel Abbès, , Monsieur **S. KERAI**, maître de conférences à l'université Abou Bakr Belkaid de Tlemcen, Monsieur **L. MERAD**, maître de conférences à l'école Supérieure en sciences appliquées de Tlemcen et Monsieur **M. SOUIER**, maître de conférences à l'école Supérieure de Management de Tlemcen, pour l'intérêt qu'ils ont porté à mes travaux en acceptant d'examiner cette thèse.*

*Je remercie Madame **S.A. BENSEGHIR** maître de conférences à l'université Abou Bakr Belkaid de Tlemcen, d'avoir accepté de participer au jury.*

*Je tiens à exprimer ma reconnaissance à Monsieur **S.M KARABERNOU**, maître de conférences à l'Ecole Nationale Supérieure de l'Electronique et de ses Applications (ENSEA) de Cergy-Pontoise, Paris, pour son aide, ses conseils précieux et ses encouragements.*

*Je remercie aussi tous les membres de l'équipe **CREDOM**, pour leurs soutiens et leurs encouragements, en particulier Nessma, Yassmine, Imane, Khalida, Amine et Mustapha.*

*Je remercie vivement mes deux amies **Dr. Boubekour Nawel** et **Dr. Belarouci Sara** pour avoir sacrifié de nombreux jours à relire cette thèse, pour leur soutien moral et leurs encouragements.*

*Ces quelques lignes ne pourront jamais exprimer la reconnaissance que j'éprouve envers tous ceux qui, de près ou de loin, ont contribué par leurs conseils, leurs aides et leurs encouragements pour l'aboutissement de ce travail.*

## ***Résumé***

Dans le domaine médical, la classification constitue le principe de base de plusieurs systèmes d'aide au diagnostic, elle utilise différentes techniques afin d'aider l'expert à établir un deuxième diagnostic pour confirmer la pathologie. Cette dernière est affirmée en temps réel en utilisant le traitement parallèle des données médicales (approche matérielle). L'étude menée dans cette thèse est portée sur l'implémentation matérielle des classifieurs neuronaux sur cible FPGA, et afin d'atteindre nos objectifs, nous avons divisé notre travail en deux phases. La première consiste à étudier le classifieur neuronal CLN de type perceptron multicouche sous l'environnement Matlab afin d'enregistrer les valeurs finales des poids synaptiques et du biais de chaque neurone, qui seront implémentées par la suite dans la deuxième phase. Cette dernière se base sur la conception matérielle du classifieur CLN, réalisée par un langage de haut niveau de description matérielle VHDL.

Afin d'avoir un compromis entre les performances relatives au temps de calcul et les besoins en ressources matérielles, notre conception est fondée sur l'emploi de deux principes de calcul différents SP (Semi-Parallèle) et PP (Parallèle-Parallèle). Ces derniers ont permis la réalisation de deux classifieurs matériels CLNH-SP et CLNH-PP, nous avons abordé le problème du dépassement de capacité en utilisant la quantification des données sur un octet. Nous avons implémenté par la suite, deux modèles de classification (SYST-SP, SYST-PP) sur le circuit FPGA permettant la reconnaissance des arythmies cardiaques en temps réel, pour de futures implantations sur des systèmes embarqués.

Ces modèles regroupent les classifieurs neuronaux déjà réalisés, en obtenant des performances acceptables pour reconnaître l'arythmie cardiaque à 200 ns pour le classifieur CLNH-SP et à 20 ns seulement pour le second CLNH-PP, et effectivement sans détection de dépassement de capacité.

### **Mots clés :**

Classifieur neuronal, Perceptron Multicouche MLP, Le langage VHDL, FPGA, Les outils de CAO, Temps réel, Arythmies cardiaques.

## *Abstract*

In the medical field, the classification constitutes the basic principle of several systems to assist in diagnosis, in order to do this, different techniques are used to help the expert in establishing a second diagnosis to confirm the pathology. However, this latter is asserted in real time using parallel processing of medical data (Hardware approach). The study carried out in this thesis focuses on the implementation of neural classifiers on FPGA target and it's based on two important phases. The first one has studied the neural classifier CLN of the multilayer Perceptron type under the environment Matlab, in order to register the final values of the synaptic weights and the bias of each neuron. These values will be used later in the hardware design. The second phase concerns the material realization of the neural classifier using a high-level language of hardware description VHDL.

In order to have a tradeoff between performance relative to computing time and material resource requirements, we have used two different calculation principles SP (Semi-Parallel) and PP (Parallel-Parallel). The latter allowed the realization of two materials classifiers CLNH-SP and CLNH-PP. we have discussed the problem of overflow that was revealed using the quantification of the data on a byte. We have implemented thereafter, two classification models (SYST-SP, SYST-PP) on the FPGA circuit allowing the recognition of cardiac arrhythmias in real time, for future implantations on embedded systems.

These models regroup the neural classifiers already achieved, obtaining acceptable performances to recognize the cardiac arrhythmia with 200 ns for the classifier CLNH-SP and only 20 ns for the second CLNH-PP, and indeed without detection of overflow.

### **Keywords:**

Neural classifier, Multilayer Perceptron MLP, VHDL language, FPGA, CAD tools, Real-time, Cardiac arrhythmias.

## الملخص

في المجال الطبي، التصنيف هو المبدأ الأساسي لعدة أنظمة في مساعدة التشخيص، إذ يستخدم تقنيات مختلفة لمساعدة الخبير في وضع تشخيص ثاني لتأكيد المرض. فإنّ هذا الأخير يثبت في الوقت الضروري باستخدام المعالجة الموازية للمعطيات الطبية (المنهج المادي). تركز الدراسة في هذه الأطروحة على التنفيذ المادي للمصنّفات المبنية على العصبونات الإصطناعية على شريحة FPGA. ولتحقيق أهدافنا، قمنا بتقسيم عملنا إلى مرحلتين. الأولى تقوم بدراسة المصنّف CLN الذي هو من نوع MLP في البيئة البرمجية MATLAB. وهذا لتسجيل القيم النهائية لمتغيّرات المصنّف، والتي سيتم استعمالها لاحقاً في المرحلة الثانية. تستند هذه الأخيرة على التصميم المادي للمصنّف CLN باستخدام لغة وصف الكيان المادي VHDL.

لإيجاد حل وسط بين النتائج المتعلقة بوقت الحساب والاحتياجات للموارد المادية، يستند تصميمنا على استخدام مبدئين مختلفين للحساب SP (شبه-موازي) و PP (موازي-موازي). مما أدى إلى بناء مصنّفين ماديّين CLNH-SP و CLNH-PP، أين تناولنا مشكلة الفائض الذي كان واضحاً باستخدام القياس الكمي للمعطيات على octet. بعد ذلك نفذنا نموذجين للتصنيف (SYST-PP، SYST-SP) على شريحة FPGA لاكتشاف الاضطرابات القلبية في الوقت الضروري، وهذا لتنفيذ مستقبلي على الأنظمة المضمنة.

تتكون هذه النماذج من المصنّفات التي صممت من قبل، و قد تحصلنا بعد تنفيذها على نتائج مقبولة لاكتشاف اضطراب قلبي في مدة 200 نانوثانية باستعمال المصنّف CLNH-SP و في مدة 20 نانوثانية فقط باستخدام المصنّف الثاني CLNH-PP و هذا بدون كشف عن أي فائض.

### الكلمات المفتاحية:

المصنّف العصبي، MLP، VHDL، FPGA، أدوات CAO، الوقت الحقيقي، الاضطرابات القلبية.

# *Table des Matières*

<b>Dédicaces</b> .....	3
<b>Remerciement</b> .....	4
<b>Résumé</b> .....	5
<b>Abstract</b> .....	6
<b>المخلص</b> .....	7
<b>Table des Matières</b> .....	8
<b>Table des Figures</b> .....	12
<b>Liste des Tableaux</b> .....	15
<b>Liste des Organigrammes</b> .....	15
<b>Notification</b> .....	16
<b>Abréviations</b> .....	17
<b>Introduction générale</b> .....	20
1. Contexte de recherche.....	21
2. Problématique, motivations et positionnement.....	22
3. Contributions.....	23
4. Organisation du manuscrit.....	24
<b>Chapitre 1. Classification neuronale des données médicales</b> .....	25
1.1. Introduction.....	26
1.2. La classification neuronale.....	26
1.2.1. Le perceptron multicouche.....	26
1.2.2. Les cartes auto-organisatrices.....	27
1.2.3. Les réseaux de neurones probabilistes.....	28
1.2.4. Les réseaux de neurones impulsionnels.....	28
1.2.5. Les classifieurs hybrides.....	29
1.3. La classification neuronale des données médicales.....	29
1.3.1. Les travaux réalisés sur les arythmies cardiaques.....	30
1.3.2. Les travaux réalisés sur les maladies de cancer.....	33
1.3.3. Les travaux réalisés sur le diabète.....	36
1.3.4. Les travaux réalisés sur les troubles neurologiques.....	36
1.4. Les avantages et les limites des approches d'implémentation.....	39
1.4.1. Approche software (logicielle).....	39
1.4.2. Approche hardware (matérielle).....	39
1.5. Conclusion.....	40



<b>Chapitre 2. Technologies et environnement de développement des FPGA.....</b>	<b>42</b>
<b>2.1. Introduction.....</b>	<b>43</b>
<b>2.2. Les circuits logiques reconfigurables FPGA.....</b>	<b>44</b>
2.2.1. Les FPGA à mémoires statiques.....	44
2.2.2. Les FPGA à mémoire Flash.....	45
2.2.3. Les FPGA à anti-fusibles.....	46
2.2.4. Les avantages et les inconvénients de différents types de FPGA.....	46
<b>2.3. Les types d'architectures des FPGA.....</b>	<b>47</b>
2.3.1. Architecture de type îlots de calcul.....	48
2.3.2. Architecture de type hiérarchique.....	48
2.3.3. Architecture de type mer de portes.....	48
<b>2.4. Les couches de FPGA.....</b>	<b>48</b>
2.4.1. Le circuit configurable.....	49
2.4.1.1. Les blocs logiques CLB.....	49
2.4.1.2. Les blocs d'entrée/sortie IOB.....	50
2.4.1.3. Les interconnexions programmables.....	51
2.4.2. Le réseau mémoire SRAM.....	54
<b>2.5. Critères de choix pour la réalisation d'un circuit logique.....</b>	<b>54</b>
2.5.1. Coût de développement et fabrication.....	54
2.5.2. Temps de développement.....	55
2.5.3. Souplesse d'utilisation.....	56
2.5.4. Taille.....	56
2.5.5. Consommation.....	56
2.5.6. Vitesse de fonctionnement.....	56
2.5.7. Capacité mémoire.....	56
<b>2.6. Notion sur les FPGA actuelles.....</b>	<b>56</b>
2.6.1. Les technologies avancées des FPGA.....	57
2.6.2. Les fabricants et parts du marché des FPGA.....	57
2.6.3. Les domaines d'application des FPGA.....	59
2.6.3.1. Domaine des systèmes embarqués.....	59
2.6.3.2. Domaine de télécommunication.....	60
2.6.3.3. Domaine médical.....	60
<b>2.7. Méthodologies de conception.....</b>	<b>60</b>
2.7.1. Méthode de conception.....	61
2.7.1.1. Le cahier des charges.....	61
2.7.1.2. La validation fonctionnelle.....	61
2.7.1.3. Le choix technologique.....	62
2.7.1.4. La synthèse.....	62
2.7.1.5. La simulation fonctionnelle.....	63
2.7.1.6. Le placement et le routage.....	63
2.7.1.7. La simulation temporelle.....	63

2.7.1.8.	La fabrication ou la programmation du circuit.....	64
2.7.2.	Les outils de développement.....	64
2.7.2.1.	Les outils de CAO (Conception Assistée par Ordinateur).....	64
2.7.2.2.	Les principaux langages de description matérielle.....	65
2.8.	Conclusion.....	66
<b>Chapitre 3. La conception matérielle du classifieur neuronal.....</b>		<b>68</b>
3.1.	Introduction.....	69
3.2.	Modèle du classifieur neuronal proposé.....	69
3.2.1.	Le perceptron multicouche.....	70
3.2.2.	L'algorithme d'apprentissage.....	71
3.2.3.	La base de données.....	72
3.2.3.1.	La base d'apprentissage.....	72
3.2.3.2.	La base de test.....	73
3.2.4.	L'arythmie ciblée.....	73
3.2.5.	La normalisation des données.....	74
3.2.6.	Apprentissage du classifieur neuronal.....	74
3.2.6.1.	L'étape d'apprentissage.....	75
3.2.6.2.	L'étape de test.....	77
3.3.	La conception matérielle du classifieur neuronal CLN.....	77
3.3.1.	Représentation des données.....	77
3.3.2.	Le principe de calcul.....	79
3.3.2.1.	Le Semi-Parallèle 'SP'.....	79
3.3.2.2.	Le Parallèle-Parallèle 'PP'.....	79
3.3.3.	L'architecture matérielle 'CLNH-SP' du classifieur neuronal.....	79
3.3.3.1.	La conception matérielle d'un neurone non complet.....	79
3.3.3.2.	Le réseau neuronal.....	82
3.3.3.3.	Le séquenceur.....	84
3.3.3.4.	Remarques.....	87
3.3.4.	La détection de dépassement de capacité.....	87
3.3.5.	L'architecture matérielle 'CLNH-PP' du classifieur neuronal.....	90
3.3.5.1.	La concrétisation d'un neurone complet.....	90
3.3.5.2.	Le classifieur matériel.....	92
3.4.	Conclusion.....	93
<b>Chapitre 4. L'implémentation du classifieur neuronal.....</b>		<b>94</b>
4.1.	Introduction.....	95
4.2.	Le flot de conception proposé.....	95
4.3.	La première implémentation du classifieur neuronal CLN.....	99
4.3.1.	Phase d'apprentissage du CLN.....	99
4.3.1.1.	Les poids synaptiques et les biais du CLN.....	99
4.3.1.2.	Les fonctions d'activation.....	100

4.3.2.	Le classifieur CLNH-SP1.....	101
4.3.3.	Résultats et discussion.....	104
4.4.	La deuxième implémentation du classifieur CLN.....	106
4.4.1.	Le classifieur CLNH-SP2.....	106
4.4.2.	Résultats et discussion.....	108
4.4.3.	Le classifieur CLNH1-SP2.....	110
4.4.3.1.	Phase d'apprentissage du classifieur CLN1.....	110
4.4.3.2.	Résultats et discussion.....	111
4.4.4.	Le classifieur CLNH2-SP2.....	113
4.4.4.1.	Phase d'apprentissage du classifieur CLN2.....	113
4.4.4.2.	Résultats et discussion.....	113
4.5.	La troisième implémentation du classifieur CLN.....	115
4.5.1.	Le classifieur CLNH-PP.....	115
4.5.2.	Le classifieur CLNH1-PP.....	118
4.5.3.	Le classifieur CLNH2-PP.....	120
4.6.	Étude comparative entre les classifieurs CLNH-SP et CLNH-PP.....	123
4.6.1.	Consommation des ressources FPGA.....	123
4.6.2.	Temps de calcul.....	126
4.7.	Conclusion.....	127
<b>Chapitre 5.</b>	<b>Classification neuronale des arythmies cardiaques à base de</b>	
<b>FPGA.....</b>	<b>.....</b>	<b>128</b>
5.1.	Introduction.....	129
5.2.	La conception d'un modèle de classification des arythmies cardiaques.....	129
5.2.1.	La reconfiguration dynamique globale du FPGA.....	133
5.2.2.	La reconfiguration dynamique partielle du FPGA.....	134
5.2.3.	Architecture en électronique câblée.....	135
5.2.3.1.	L'architecture matérielle du modèle SYST-SP.....	135
5.2.3.2.	L'architecture matérielle du modèle SYST-PP.....	137
5.2.3.3.	La synthèse des deux modèles de classification.....	140
5.3.	Travaux connexes.....	140
5.4.	Conclusion.....	144
<b>Conclusion générale et perspectives.....</b>	<b>.....</b>	<b>145</b>
<b>Annexes.....</b>	<b>.....</b>	<b>148</b>
<b>Références bibliographiques.....</b>	<b>.....</b>	<b>157</b>
<b>Productions scientifiques.....</b>	<b>.....</b>	<b>171</b>

# Table des figures

## Chapitre 1

<b>Figure 1.1.</b> Un réseau perceptron multicouche.....	27
<b>Figure 1.2.</b> L'architecture d'une carte de Kohonen.....	27
<b>Figure 1.3.</b> Le réseau neuronal temporel.....	29

## Chapitre 2

<b>Figure 2.1.</b> Classification des systèmes numériques de haute densité.....	43
<b>Figure 2.2.</b> FPGA XILINX.....	44
<b>Figure 2.3.</b> Cellule SRAM.....	45
<b>Figure 2.4.</b> MOS à grille flottante.....	46
<b>Figure 2.5.</b> La technique des anti-fusibles.....	46
<b>Figure 2.6.</b> Le concept d'un FPGA du type SRAM [XILINX_2000].....	48
<b>Figure 2.7.</b> Structure interne du circuit configurable [XILINX_2000].....	49
<b>Figure 2.8.</b> Un bloc CLB ( <i>Configurable Logic Blocs</i> ) [XILINX_2000].....	50
<b>Figure 2.9.</b> Schéma d'une cellule logique (XC4000 de XILINX) [XILINX_1999].....	50
<b>Figure 2.10.</b> Schéma d'un bloc d'entrée/sortie (SPARTAN) [XILINX_2008].....	51
<b>Figure 2.11.</b> Connexion à usage général.....	52
<b>Figure 2.12.</b> Concept d'une matrice de commutation.....	53
<b>Figure 2.13.</b> Les interconnexions directes.....	53
<b>Figure 2.14.</b> Les longues lignes.....	53
<b>Figure 2.15.</b> Coût des FPGA par rapport aux ASIC [KARABERNOU_2009].....	55
<b>Figure 2.16.</b> Temps de conception [KARABERNOU_2016].....	55
<b>Figure 2.17.</b> L'intégration dans les circuits logiques.....	56
<b>Figure 2.18.</b> Évolution des technologies d'intégrations des FPGA.....	57
<b>Figure 2.19.</b> Les principaux fabricants des circuits FPGA.....	58
<b>Figure 2.20.</b> Le marché des FPGA.....	58
<b>Figure 2.21.</b> Simulation par un banc de test.....	63
<b>Figure 2.22.</b> L'utilité d'un outil de développement.....	64

## Chapitre 3

<b>Figure 3.1.</b> Structure d'un neurone formel.....	70
<b>Figure 3.2.</b> Structure d'un perceptron à deux couches.....	70
<b>Figure 3.3.</b> Apprentissage d'un perceptron multicouche par rétro-propagation du gradient....	71
<b>Figure 3.5.</b> ECG d'un sujet sain [CARRE_1984].....	73
<b>Figure 3.5.</b> ECG d'un sujet qui présente des extrasystoles ventriculaires [CARRE_1984]....	74

<b>Figure 3.6.</b> Visualisation du réseau réalisé sur la boîte à outils.....	75
<b>Figure 3.7.</b> Les meilleures performances de validation.....	76
<b>Figure 3.8.</b> Les résultats obtenus lors d'une classification.....	77
<b>Figure 3.9.</b> Représentation d'un nombre en binaire.....	78
<b>Figure 3.10.</b> Représentation des nombres signés et non signés sur 8 bits.....	78
<b>Figure 3.11.</b> Architecture matérielle d'un seul neurone.....	80
<b>Figure 3.12.</b> La gestion des mémoires ROM dans le neurone.	
<b>(a).</b> Rom d'un neurone de la couche cachée.....	81
<b>(b).</b> Rom du neurone de la couche de sortie.....	81
<b>Figure 3.13.</b> Architecture matérielle du réseau neuronal.....	83
<b>Figure 3.14.</b> La fonction Tangente Hyperbolique Sigmoidale.....	84
<b>Figure 3.15.</b> Bloc du Séquenceur.....	85
<b>Figure 3.16.</b> La gestion des signaux de commande dans le séquenceur.....	86
<b>Figure 3.17.</b> La nouvelle architecture matérielle du Neurone_NC.....	88
<b>Figure 3.18.</b> Le fonctionnement d'une bascule lors de la détection du dépassement de capacité.....	89
<b>Figure 3.19.</b> Architecture matérielle de la couche cachée du classifieur neuronal CLN.....	90
<b>Figure 3.20.</b> Architecture matérielle d'un neurone de la couche cachée.....	91
<b>Figure 3.21.</b> Architecture matérielle 'CLNH-PP' du classifieur neuronal.....	92

## *Chapitre 4*

<b>Figure 4.1.</b> Schéma général du flot de conception.....	96
<b>Figure 4.2.</b> L'environnement ModelSimSe 6.2c.....	97
<b>Figure 4.3.</b> Les fichiers des stimules et des résultats du CLNH.	
<b>(a).</b> Fichier contenant les paramètres d'entrée.....	97
<b>(b).</b> Fichiers contenant les classes et les dépassements de capacité.....	98
<b>Figure 4.4.</b> L'environnement ISE Design 14.7.....	99
<b>Figure 4.5.</b> Les fonctions d'activation des deux couches du CLN.	
<b>(a).</b> Tansig (6x) de la couche cachée.....	101
<b>(b).</b> Tansig (4x) de la couche de sortie.....	101
<b>Figure 4.6.</b> La simulation du bloc Rom sous l'environnement ModelSim.....	102
<b>Figure 4.7.</b> La simulation du bloc multiplieur sous l'environnement ModelSim.....	102
<b>Figure 4.8.</b> La simulation du bloc additionneur sous ModelSim.....	102
<b>Figure 4.9.</b> La simulation du bloc de séquenceur sous ModelSim.....	103
<b>Figure 4.10.</b> Le taux de classification obtenu des classifieurs CLN et CLNH-SP1.....	105
<b>Figure 4.11.</b> La simulation du nouveau additionneur sous ModelSim.....	106
<b>Figure 4.12.</b> La simulation d'un Neurone_NC de la couche cachée sous ModelSim.....	107
<b>Figure 4.13.</b> La simulation du bloc de la couche cachée sous ModelSim.....	107
<b>Figure 4.14.</b> La simulation du bloc comparateur sous ModelSim.....	108
<b>Figure 4.15.</b> Le taux de classification obtenu des classifieurs CLN et CLNH-SP2.....	109
<b>Figure 4.16.</b> La simulation du classifieur CLNH-SP2 sous ModelSim.....	109
<b>Figure 4.17.</b> Le taux de classification obtenu des classifieurs CLN1 et CLNH1-SP2.....	111

<b>Figure 4.18.</b> La simulation du classifieur CLNH1-SP2 sous ModelSim.	
<b>(a).</b> Le test du 1 <sup>er</sup> cycle cardiaque du sujet 119.....	112
<b>(b).</b> Le test du 88 <sup>eme</sup> cycle cardiaque du sujet 119.....	112
<b>Figure 4.19.</b> Le taux de classification obtenu des classifieurs CLN2 et CLNH2-SP2.....	114
<b>Figure 4.20.</b> La simulation du classifieur CLNH2-SP2 du sujet 119 sous l'environnement ModelSim.....	115
<b>Figure 4.21.</b> La simulation du bloc Neurone sous l'environnement ModelSim.....	116
<b>Figure 4.22.</b> Le taux de classification obtenu des classifieurs CLN et CLNH-PP.....	117
<b>Figure 4.23.</b> La simulation du classifieur CLNH-PP sous ModelSim.....	117
<b>Figure 4.24.</b> Le taux de classification obtenu des classifieurs CLN1 et CLNH1-PP.....	119
<b>Figure 4.25.</b> La simulation du classifieur CLNH1-PP sous ModelSim.	
<b>(a).</b> Le test des 6 premiers cycles cardiaques du sujet 106.....	119
<b>(b).</b> Le test des cycles cardiaques (92, 93, 94, 95, 96, 97, 98) du sujet 106...	120
<b>Figure 4.26.</b> Le taux de classification obtenu des classifieurs CLN2 et CLNH2-PP.....	121
<b>Figure 4.27.</b> La simulation du classifieur CLNH2-PP sous l'environnement ModelSim.....	122
<b>Figure 4.28.</b> La structure RTL du classifieur CLNH-SP2.....	125
<b>Figure 4.29.</b> La structure RTL du classifieur CLNH-PP.....	125
<b>Figure 4.30.</b> Le temps d'exécution des classifieurs CLNH2-SP2 et CLNH2-PP.....	126

## *Chapitre 5*

<b>Figure 5.1.</b> Les quatre configurations du modèle de classification SYST-SP.....	131
<b>Figure 5.2.</b> L'architecture matérielle du modèle SYST-SP.....	136
<b>Figure 5.3.</b> L'architecture matérielle du modèle SYST-PP.....	139

## *Annexes*

<b>Figure A.1.</b> Le signal ECG.....	148
<b>Figure A.2.</b> Système d'IRM à 3 Tesla.....	151
<b>Figure A.3.</b> Principaux rythmes de l'EEG.....	152

## ***Liste des Tableaux***

<b>Tableau 1.1.</b> Comparaison entre les deux approches d'implémentation.....	40
<b>Tableau 2.1.</b> Comparaison entre les trois types de FPGA.....	47
<b>Tableau 3.1.</b> Les enregistrements utilisés pour la conception de la base d'apprentissage.....	72
<b>Tableau 3.2.</b> Les enregistrements utilisés pour la phase de test.....	73
<b>Tableau 3.3.</b> Représentation des données.....	79
<b>Tableau 3.4.</b> Le principe de l'algorithme de BOOTH.....	82
<b>Tableau 3.5.</b> Le débordement de calcul ( <i>overflow</i> ).....	88
<b>Tableau 4.1.</b> Les valeurs finales des poids synaptiques et des biais du CLN.....	99
<b>Tableau 4.2.</b> Les performances des classifieurs CLN et CLNH-SP1.....	104
<b>Tableau 4.3.</b> Le calcul de la sensibilité et de la spécificité des classifieurs CLN et CLNH-SP1.....	105
<b>Tableau 4.4.</b> Les performances des classifieurs CLN et CLNH-SP2.....	108
<b>Tableau 4.5.</b> Les valeurs finales des poids synaptiques et des biais du CLN1.....	110
<b>Tableau 4.6.</b> Les performances des classifieurs CLN1 et CLNH1-SP2.....	111
<b>Tableau 4.7.</b> Les valeurs finales des poids synaptiques et des biais du CLN2.....	113
<b>Tableau 4.8.</b> Les performances des classifieurs CLN2 et CLNH2-SP2.....	113
<b>Tableau 4.9.</b> Le calcul de la sensibilité et de la spécificité des classifieurs CLN2 et CLNH2-SP2.....	114
<b>Tableau 4.10.</b> Les performances des classifieurs CLN et CLNH-PP.....	116
<b>Tableau 4.11.</b> Les performances des classifieurs CLN1 et CLNH1-PP.....	118
<b>Tableau 4.12.</b> Les performances des classifieurs CLN2 et CLNH2-PP.....	121
<b>Tableau 4.13.</b> Le calcul de la sensibilité et de la spécificité des classifieurs CLN2 et CLNH2-PP.....	122
<b>Tableau 4.14.</b> Consommation des ressources matérielles.....	123
<b>Tableau 4.15.</b> Le temps d'exécution des classifieurs CLNH2-SP2 et CLNH2-PP.....	126
<b>Tableau 5.1.</b> Consommation des ressources matérielles des deux modèles de classification.....	140
<b>Tableau 5.2.</b> Présentation des différentes méthodologies mises en œuvre sur FPGA.....	143

## ***Liste des organigrammes :***

<b>Organigramme 2.1.</b> La conception des circuits à hautes densités.....	61
<b>Organigramme 3.1.</b> Les différentes étapes pour l'apprentissage du classifieur neuronal CLN.....	76
<b>Organigramme 5.1.</b> Le fonctionnement du modèle de classification SYST-SP.....	132
<b>Organigramme 5.2.</b> La reconfiguration dynamique du modèle SYST-SP.....	134

## *Notation*

<b>A :</b>	Nombre quantifié sur un octet.
<b>a(i) :</b>	La somme des entrées pondérées.
<b>B :</b>	Nombre quantifié sur un octet.
<b>b(i) :</b>	Les valeurs des biais du réseau de neurones.
<b>Energie :</b>	L'énergie moyenne du complexe QRS au carré.
<b>F :</b>	La fonction d'activation du réseau de neurones.
<b>K :</b>	Le coefficient est utilisé pour la minimisation des valeurs des paramètres d'entrée.
<b>N :</b>	La classe négative qui représente les battements normaux.
<b>n :</b>	Le nombre d'entrées du réseau de neurones.
<b>p(j) :</b>	Le vecteur des entrées du réseau de neurones.
<b>Peak :</b>	L'amplitude de peak de l'onde R.
<b>PP :</b>	L'amplitude (peak to peak) de l'onde R jusqu'à l'onde S.
<b>QRS :</b>	La durée du complexe QRS.
<b>RRp :</b>	La distance entre l'onde R actuelle et l'onde R précédente.
<b>RRs :</b>	La distance entre l'onde R actuelle et l'onde R suivante.
<b>RRs/RRp :</b>	Le rapport entre la distance RR suivante sur la précédente.
<b>S :</b>	La somme des nombres A et B, elle est quantifiée sur un octet.
<b>T(i) :</b>	Le vecteur de sortie du réseau de neurones.
<b>V :</b>	La classe positive qui représente les battements des extrasystoles ventriculaires ESV.
<b>W(i, j) :</b>	La matrice des poids synaptiques du réseau de neurones..
<b><math>\alpha</math> :</b>	Un pas ( $\alpha = 0.2$ ) est utilisé pour augmenter la valeur du coefficient K par ce pas, et cela à partir de $1 + \alpha$ .
<b><math>\beta</math> :</b>	Le nombre d'accumulations effectués dans les couches du réseau de neurones.
<b><math>\lambda</math> :</b>	Le coefficient est utilisé pour normaliser les valeurs des poids synaptiques et des biais.



## *Abréviations*

<b>Add :</b>	Un bloc d'additionneur.
<b>ANFIS:</b>	An adaptive neuro-fuzzy inference system.
<b>ANN:</b>	Artificial Neural Network.
<b>ASIC:</b>	Application Specific Integrated Circuit.
<b>Bloc_Valeurs :</b>	Les blocs des valeurs des poids synaptiques et des biais dans le classifieur CLNH-PP.
<b>Bloc2_Valeurs :</b>	Les blocs des valeurs des poids synaptiques et des biais dans le classifieur CLNH1-PP.
<b>Bloc4_Valeurs :</b>	Les blocs des valeurs des poids synaptiques et des biais dans le classifieur CLNH2-PP.
<b>Bloc8_Valeurs :</b>	Les blocs des valeurs des poids synaptiques et des biais dans le classifieur CLNH3-PP.
<b>CAO:</b>	Conception Assistée par Ordinateur.
<b>CLB:</b>	Configurable Logic Blocks.
<b>CLN :</b>	Classifieur Neuronal de type perceptron Multicouche (7-3-1).
<b>CLN1 :</b>	Classifieur Neuronal qui utilise les valeurs des paramètres d'entrée divisées par deux (/2).
<b>CLN2 :</b>	Classifieur Neuronal qui utilise les valeurs des paramètres d'entrée divisées par deux (/4).
<b>CLNH1-PP:</b>	Classifieur neuronal matériel qui utilise le (PP) et les valeurs des paramètres d'entrée divisées par deux (/2).
<b>CLNH1-SP2 :</b>	Classifieur neuronal matériel qui utilise le (SP) et les valeurs des paramètres d'entrée divisées par deux (/2).
<b>CLNH2-PP:</b>	Classifieur neuronal matériel qui utilise (PP) et les valeurs des paramètres d'entrée divisées par deux (/4).
<b>CLNH2-SP2 :</b>	Classifieur neuronal matériel qui utilise le (SP) et les valeurs des paramètres d'entrée par quatre (/4).
<b>CLNH3-PP :</b>	Classifieur neuronal matériel qui utilise (PP) et les valeurs des paramètres d'entrée divisées par deux (/8).
<b>CLNH3-SP2 :</b>	Classifieur neuronal matériel qui utilise le (SP) et les valeurs des paramètres d'entrée divisées par huit (/8).
<b>CLNH-PP :</b>	Classifieur neuronal matériel utilisant le principe de calcul 'Parallèle-Parallèle'.
<b>CLNH-SP :</b>	Classifieur neuronal matériel utilisant le principe de calcul 'Semi-Parallèle'.
<b>CLNH-SP1:</b>	Classifieur neuronal matériel sans détection de dépassement de capacité.
<b>CLNH-SP2 :</b>	Classifieur neuronal matériel avec détection de dépassement de capacité.

<b>CMOS:</b>	Complementary Metal-Oxide Silicon.
<b>CPLD:</b>	Complex Logic Programmable Device.
<b>ECG:</b>	Électrocardiogramme.
<b>EEG:</b>	Électroencéphalogramme.
<b>EEPROM:</b>	Electrically Erasable Programmable Read-Only Memory.
<b>ESV :</b>	Extrasystoles ventriculaires.
<b>FN:</b>	Faux Négatifs.
<b>Fonct_Tansig :</b>	Les blocs de la fonction d'activation dans les classifieurs CLNH-SP2 et CLNH-PP.
<b>Fonct2_Tansig :</b>	Les blocs de la fonction d'activation dans les classifieurs CLNH1-SP2 et CLNH1-PP.
<b>Fonct4_Tansig :</b>	Les blocs de la fonction d'activation dans les classifieurs CLNH2-SP2 et CLNH2-PP.
<b>Fonct8_Tansig :</b>	Les blocs de la fonction d'activation dans les classifieurs CLNH3-SP2 et CLNH3-PP.
<b>FP:</b>	Faux positifs.
<b>FPGA:</b>	Field Programmable Gate Arrays.
<b>HDL:</b>	Hardware Description Language.
<b>IOB:</b>	Input/Output Blocks.
<b>IRM :</b>	Imagerie par Résonance Magnétique.
<b>LCA:</b>	Logic Cell Array.
<b>LSI:</b>	Large Scale Integration.
<b>LUT:</b>	Look Up Table.
<b>MIT-BIH:</b>	Massachusetts Institute of Technology/ Beth Israel Hospital.
<b>MLP:</b>	Multi Layer Perceptron.
<b>MOS:</b>	Metal oxyd semiconductor.
<b>MSB:</b>	Most Significant Bit.
<b>MSI:</b>	Medium Scale Integration.
<b>Mult_Booth:</b>	Un bloc de multiplieur qui utilise l'algorithme de Booth.
<b>NbR:</b>	Le nombre de cas reconnus correctement.
<b>Neurone :</b>	Un bloc d'un neurone complet
<b>Neurone_NC:</b>	Un bloc d'un neurone non complet.
<b>Ovf :</b>	Le nombre de dépassement de capacité détecté (OverFlows).
<b>PAL:</b>	Programmable Array Logic.
<b>PLD:</b>	Programmable Logic Device.
<b>PNN :</b>	Probabilistic Neural Network.
<b>Poids_Rom :</b>	Les blocs de Rom dans le classifieur CLNH-SP2.
<b>Poids2_Rom :</b>	Les blocs de Rom dans le classifieur CLNH1-SP2.
<b>Poids4_Rom :</b>	Les blocs de Rom dans le classifieur CLNH2-SP2.
<b>Poids8_Rom :</b>	Les blocs de Rom dans le classifieur CLNH3-SP2.
<b>PP :</b>	Principe de calcul utilisé 'Parallèle-Parallèle'.
<b>RAM:</b>	Random Access Memory.
<b>Registre_Biais :</b>	Les blocs de registre dans le classifieur CLNH-SP2.
<b>Registre2_Biais :</b>	Les blocs de registre dans le classifieur CLNH1-SP2.

<b>Registre4_Biais :</b>	Les blocs de registre dans le classifieur CLNH2-SP2.
<b>Registre8_Biais :</b>	Les blocs de registre dans le classifieur CLNH3-SP2.
<b>ROM:</b>	Read Only Memory.
<b>RTL:</b>	Register Transfer Logic.
<b>RTR:</b>	Run Time Reconfiguration
<b>Se:</b>	La sensibilité.
<b>SNN:</b>	Spiking Neural Network.
<b>SoC:</b>	System on Chip.
<b>SOM:</b>	Self organizing map.
<b>SoPC:</b>	System on a Programmable Chip.
<b>Sp:</b>	La spécificité.
<b>SP:</b>	Principe de calcul utilisé ‘Semi-Parallèle’.
<b>SPLD:</b>	Sample Logic Programmable Device.
<b>SRAM:</b>	Static Random Access Memory.
<b>SSI:</b>	Small Scale Integration.
<b>SYST-PP :</b>	Modèle matériel de classification, utilisant les classifieurs CLNH-PP.
<b>SYST-SP :</b>	Modèle matériel de classification, utilisant les classifieurs CLNH-SP.
<b>TC:</b>	Le taux de reconnaissance correct.
<b>TN:</b>	Vrai Négatifs.
<b>TP:</b>	Vrai Positifs.
<b>TTL:</b>	Transistor Transistor Logic.
<b>VHDL:</b>	Very High Speed Integrated Circuit Hardware Description Language.
<b>VLSI:</b>	Very Large Scale Integration.
<b>WNN:</b>	Wavelet Neural Network.



***Introduction Générale.***

## **1. Contexte de recherche :**

Aujourd'hui, les connaissances médicales et l'identification exacte des maladies sont devenues de plus en plus nombreuses et complexes, ce qui rend la tâche du médecin de plus en plus difficile car la mémoire humaine est limitée et le médecin ne peut avoir en tête l'ensemble du savoir médical. Pour remédier à ce problème, les qualités des nouvelles techniques sont devenues comme des solutions éventuelles à cette difficulté, d'où la naissance des systèmes automatisés d'aide au diagnostic [CLERET\_2001].

La plupart de ces systèmes sont basés sur le principe de classification, cette dernière utilise des algorithmes et des techniques dont le but principal est d'aider les experts dans le domaine médical dans leurs diagnostics. Dans la littérature, nous trouvons plusieurs techniques appliquant les approches de l'intelligence artificielle dans la tâche de la classification, telles que l'approche bayésienne [RAPHAËL\_2010], les Supports Vectors Machines (SVM) [VANITHA\_2011], les réseaux de neurones artificiels [JADHAV\_2012] et d'autres.

L'utilisation des techniques de réseaux neuronaux montre un grand potentiel dans le domaine du diagnostic médical, ils constituent un modèle computationnel puissant et ils permettent de mettre au point les systèmes de classification. Les réseaux de neurones ont des propriétés remarquables comme l'apprentissage à partir des exemples, la généralisation, la capacité d'adaptation et la robustesse face au bruit. Ils sont intéressants non seulement à cause de leur polyvalence mais aussi pour la simplicité de leur mise en œuvre.

La structure parallèle des réseaux neuronaux rend le calcul très rapide et efficace pour la classification des données, en exigeant une puissance de calcul énorme [PREVOTET\_2003]. Par conséquent, l'utilisation de ces classifieurs dans les systèmes informatiques est loin d'offrir des performances intéressantes en temps pour une détection précoce d'une anomalie, et cela est dû au traitement séquentiel de ces systèmes. En revanche, le parallélisme des réseaux neuronaux est très bien adapté pour l'implémentation matérielle sur des circuits de haute densité d'intégration (VLSI), soit en utilisant les techniques de conceptions rapides et modifiables (reprogrammable) offertes par les circuits du type FPGA, ou en réalisant l'application d'une manière plus optimisée sur un circuit du type ASIC.

Les inconvénients majeurs d'une conception optimisée résident dans le fait du passage obligatoire chez le fondeur, ce qui implique des frais de développement élevés du circuit, et il s'avère nécessaire d'être rigoureux lors de la phase de développement, de telle sorte que le circuit prototype fonctionné dès les premiers essais. Quant à l'implémentation sur un circuit reprogrammable du type FPGA, la conception et la réalisation des architectures neuronales sont obtenues avec relativement un faible investissement de développement [HAITHAM\_2010], [RAFID\_2007], [EL MOUKHLIS\_2014].

Les cellules du FPGA étant indépendantes les unes des autres, d'où elles peuvent très bien effectuer leurs calculs en parallèle en fournissant une puissance de calcul importante, ce qui les rend rapides, en permettant d'intégrer différents algorithmes [PINJARE\_2012]. Ces derniers sont considérés comme une séquence d'instructions, qui peuvent être analysées pour

déterminer un parallélisme possible. La possibilité de reconfiguration sur les circuits FPGA permet une minimisation et une adaptation spécifique à chaque algorithme [SAHIN\_2006].

## **2. Problématique, motivations et positionnement :**

Dans le domaine médical, l'expert est confronté au traitement d'une grande quantité de données médicales afin d'établir son diagnostic, et aussi le traitement de ces données en utilisant un mode séquentiel est confronté à différents problèmes, nous citons en particulier le temps important pour réaliser le diagnostic à cause de l'accès fréquent au mémoire du système.

Afin de pallier ce type de problème, une méthode et des outils adaptés à une implémentation matérielle sont nécessaires. Le but principal d'une telle méthode est de faciliter la tâche à l'expert en lui fournissant un temps réduit pour la reconnaissance des pathologies, par exemple, il faut avoir le diagnostic en temps réel d'un cas critique d'un malade sous soins intensifs où l'intervention immédiate est plus que nécessaire pour la survie du malade. Ainsi, cette méthode permet d'aider l'expert à localiser les données pertinentes, et d'attirer l'attention de l'expert sur les points pertinents du malade par un système automatisé d'alerte.

Le but principal de cette thèse est de réaliser une conception matérielle d'un classifieur neuronal des données médicales afin d'être implémentée sur un circuit reprogrammable du type FPGA pour la détection d'une pathologie en temps réel. Notre étude a été orientée sur les arythmies cardiaques vues qu'elles constituent l'une des principales causes de mortalité dans le monde, d'où la nécessité de détecter ces anomalies au moment de leurs apparitions. Parmi les examens les plus couramment effectués pour leur détection, nous trouvons l'électrocardiogramme (ECG) qui possède une grande quantité de données dont l'analyse manuelle est difficilement envisageable. Par conséquent, les systèmes intelligents capables d'aider les médecins à faire le diagnostic et le suivi des malades deviennent indispensables. Divers travaux de recherche concernant la classification automatique des signaux ECG ont été proposés dans la littérature ces dernières années. En particulier, les réseaux de neurones [ZHOU\_2003], [ROSARIA\_1998], [MARK\_1997], la logique floue [MESSAOUUD\_2005], les Supports Vectors Machines (SVM) [ZIDELMAL\_2012] et l'hybridation des différentes méthodes [HENDEL\_2010], [GOTHWAL\_2011].

La matérialisation de ces classifieurs ouvre une autre piste dans la reconnaissance des arythmies cardiaques, car elle permet d'avoir la détection des pathologies en temps réel en utilisant des circuits à haute densité d'intégration dédiés à l'application (ASIC) ou des circuits reconfigurables (FPGA). Notre choix est orienté sur ces derniers, car ils proposent un compromis intéressant entre les flexibilités de processeur et les performances de la logique câblée, ils sont trop adaptés à l'évolutivité des systèmes, leurs fabrications et leurs tests sont des étapes courtes et peu coûteuses par rapport à un circuit ASIC. En revanche, ils ont une densité d'intégration limitée et une consommation d'énergie importante. Les circuits FPGA sont également une véritable opportunité pour obtenir un prototypage rapide des réseaux de neurones au sein d'une application de classification, tout en permettant ensuite d'aboutir à la

conception d'un circuit intégré dédié ASIC. Plusieurs travaux présentés dernièrement proposent l'implémentation matérielle sur les circuits FPGA des systèmes d'aide au diagnostic pour la classification des arythmies cardiaques [ARMATO\_2009], [JEWAJINDA\_2010], [OZDEMIR\_2011], [JEMBULA\_2013], [EGILA\_2015], [BABA\_2016].

### **3. Contributions :**

La contribution principale de ce travail est la proposition d'une approche de développement pour la conception matérielle d'un classifieur neuronal afin de reconnaître les arythmies cardiaques, particulièrement les extrasystoles ventriculaires (ESV) en temps réel. Cette concrétisation est réalisée à l'aide d'un langage de haut niveau de description matérielle VHDL, en vérifiant le bon fonctionnement du classifieur par l'outil de simulation ModelSim 6.2c de Mentor Graphics, puis nous effectuons la synthèse du circuit sous l'environnement ISE Design 14.7 de la société Xilinx.

Dans le cadre de cette thèse, nous avons utilisé un perceptron multicouche (MLP), puisqu'il est l'un des réseaux de neurones le plus répandu et le plus simple à mettre en œuvre. Le développement du MLP est réalisé sous l'environnement Matlab en utilisant un algorithme d'apprentissage supervisé « la rétro-propagation ». Ensuite, les coefficients des poids synaptiques et du biais finaux de chaque neurone sont enregistrées afin de les utiliser dans l'architecture matérielle du classifieur neuronal. Ceci veut dire que la conception du réseau neuronal est effectuée indépendamment du matériel (hors du circuit FPGA), puisque l'apprentissage sur circuit est souvent considéré comme difficile et inutile, même il est habituellement la cause d'une perte d'efficacité dans une implémentation matérielle, car il requiert des opérateurs spécifiques et une précision accrue. C'est pourquoi l'apprentissage primitif sur ordinateur est choisi lorsqu'aucun apprentissage continu n'est nécessaire.

Dans la réalisation d'un projet, un compromis est essentiel entre les principaux enjeux tels que la surface du circuit, la vitesse d'exécution, la consommation et la précision (cahier des charges). Dans notre travail, l'architecture matérielle du classifieur neuronal (CLN) contient différents blocs (Multiplieurs, Additionneurs, registres et etc.) qui permettent d'effectuer le calcul des données (paramètres d'entrée, poids synaptiques et biais) quantifiées en virgule fixe sur un octet dans un intervalle [-1, +1].

Cette thèse propose des solutions de conception matérielle équilibrée pour la reconnaissance des ESVs. Dans un premier temps, nous réalisons la première solution qui est centrée sur l'utilisation d'un principe de calcul Semi-Parallèle (SP). Ce dernier consiste à transmettre l'information séquentiellement d'une couche vers l'autre, en effectuant un calcul parallèle dans la même couche du classifieur matériel (CLNH-SP). Dans cette implémentation, l'occupation de ressources dans le FPGA est minimale au détriment du temps d'exécution. Lors de la simulation du classifieur CLNH-SP, nous remarquons un débordement dans le calcul au niveau des additionneurs, qui est due au choix de la quantification des données. Pour remédier à ce problème, nous réalisons une nouvelle conception matérielle du

neurone, permettant de détecter le dépassement de capacité, et afin d'améliorer nos résultats, nous appliquons plusieurs protocoles de calculs expérimentaux.

Dans un deuxième temps, nous proposons la nouvelle architecture matérielle du CLN, en adoptant un autre principe de calcul Parallèle-Parallèle (PP). Ce dernier consiste à transmettre l'information d'une couche vers l'autre en parallèle, en effectuant un calcul parallèle dans la même couche du classifieur matériel (CLNH-PP). Ce type d'implantation peut vite s'avérer comme un consommateur de ressources, en fournissant un temps d'exécution rapide. Enfin, nous présentons une technique permettant la réalisation des modèles de classification (SYST-SP, SYST-PP) en vue d'une future exploitation dans un système embarqué capable de détecter les arythmies cardiaques en temps réel.

#### **4. Organisation du manuscrit :**

Cette thèse est subdivisée en cinq chapitres :

Le premier chapitre présente une revue de littérature de ces dernières années sur la classification neuronale des données médicales. L'objectif visé par ce chapitre est de mettre la lumière sur les avantages et les inconvénients des deux approches de classification (l'implémentation *Software* et *Hardware*).

Dans le deuxième chapitre, nous découvrons la technologie et l'environnement de développement des circuits reconfigurables FPGA, ainsi que les domaines d'application de ces derniers.

Le troisième chapitre constitue le cœur de cette thèse, d'où nous présentons le développement du classifieur neuronal sous l'environnement Matlab en montrant ses différentes parties, puis nous exposerons l'architecture matérielle globale du classifieur en adoptant deux principes de calcul différents. Ensuite, nous détaillerons les différents blocs de nos architectures et leurs rôles respectifs.

Le quatrième chapitre est consacré aux différentes implémentations du classifieur neuronal ainsi que les performances obtenues, puis nous proposons une étude comparative entre les deux architectures matérielles en termes de ressources et de temps de calcul.

Dans le cinquième chapitre, nous présentons l'architecture générale des deux modèles de classification des arythmies cardiaques en temps réel, en proposant un panorama des différentes stratégies et techniques proposées dans le cadre de cette reconnaissance. Enfin, nous terminons ce chapitre par la présentation des travaux connexes.

Nous terminons cette thèse avec une conclusion générale et des perspectives.





*Chapitre 1*

*Classification Neuronale des  
Données Médicales.*

## 1.1. Introduction :

La classification est un domaine de recherche qui a été développé depuis plusieurs années, son objectif est de scinder l'échantillon de données en groupes d'observations homogènes, chaque groupe étant bien différencié des autres. Une obtention par la suite des sections à l'intérieur des groupes principaux, qui vont être subdivisés. Le but final, c'est avoir une distinction de plus en plus fine, sur l'ensemble d'observations initiales [ROUX\_1985].

Dans l'analyse des signaux physiologiques, les réseaux de neurones artificiels sont fréquemment utilisés pour la classification, en raison de leur capacité à résoudre des problèmes non linéaires avec beaucoup de succès. Pour une tâche de classification, le réseau neuronal artificiel cherche une structure de réseau avec un ensemble de paramètres à l'aide d'un algorithme d'apprentissage.

Dans ce chapitre, nous allons définir les réseaux de neurones utilisés dans la littérature. Ces derniers vont être exposés via deux approches d'implémentation (*Software* et *Hardware*), en montrant l'intérêt des classifieurs neuronaux dans la détection des différentes pathologies. Enfin, nous allons présenter les avantages et les inconvénients de chaque approche d'implémentation.

## 1.2. La classification neuronale :

Les réseaux de neurones artificiels ANNs (*Artificial Neural Networks*) sont apparus comme des outils intéressants dans divers domaines industriels et de recherche. Ils sont capables de résoudre les problèmes non linéaires et complexes comme la reconnaissance des formes, la classification ou l'optimisation.

Un réseau neuronal artificiel est un modèle mathématique de calcul qui s'inspire de la structure des aspects fonctionnels des réseaux de neurones biologiques. L'apprentissage des ANNs est réalisé grâce à des algorithmes, ces derniers sont développés selon certaines règles d'apprentissage spéciaux.

Nous citons quelques types de réseaux neuronaux utilisés dans les travaux présentés dans l'état de l'art.

### 1.2.1. Le perceptron multicouche :

Le perceptron multicouche MLP (*Multi Layer Perceptron*) est un modèle non linéaire, le plus utilisé dans les réseaux de neurones. Il peut contenir deux ou plusieurs couches (Figure 1.1). La couche d'entrée est la première couche dont le nombre de neurones égal au nombre de caractéristiques spécifiques sélectionnées. La couche de sortie est la dernière couche qui détermine la sortie, le nombre de neurones dans cette couche dépend du nombre de classes désirées. Les couches intermédiaires sont ajoutées afin de renforcer la capacité d'apprentissage du réseau.

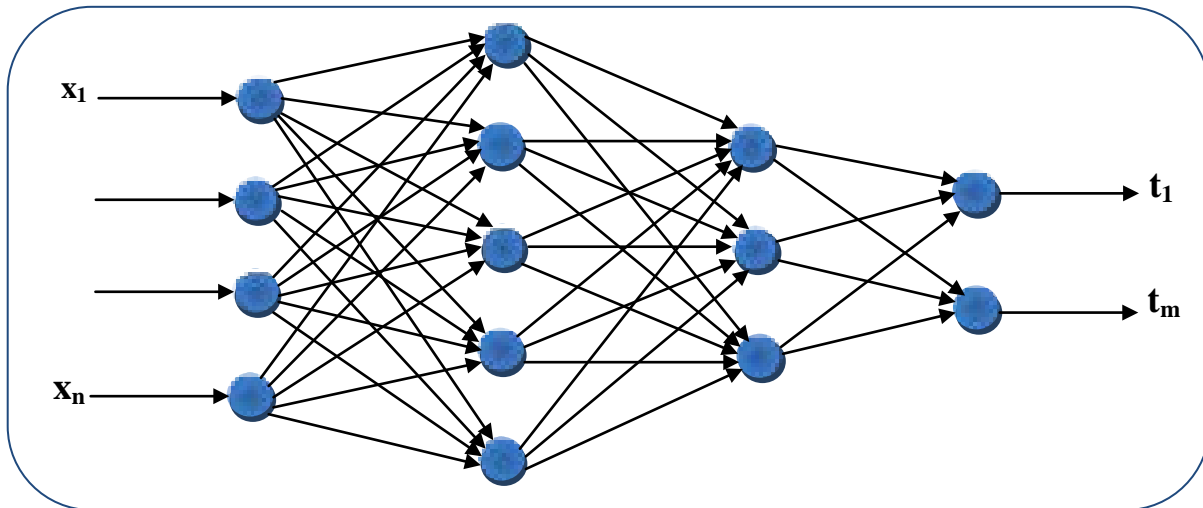


Figure 1.1. Un réseau perceptron multicouche.

### 1.2.2. Les cartes auto-organisatrices :

Les cartes auto-organisatrices SOM (self organizing maps), sont des cartes topologiques qui forment une classe de réseau de neurones artificiels fondée sur des méthodes d'apprentissage non supervisées. Le concept de ces cartes a été développé par le statisticien Kohonen en 1984 [WIKIPEDIA\_09/2016]. Elles sont utilisées pour étudier la répartition de données dans un espace à grande dimension. En pratique, cette cartographie peut servir à réaliser des tâches de discrétisation, quantification vectorielle ou classification.

La carte de Kohonen se compose de deux couches (Figure 1.2), une couche d'entrée et l'autre de sortie. Cette dernière est appelée aussi couche compétitive en deux dimensions. La connexion entre les neurones d'entrée et de sortie se fait par des poids  $W_{ij}$ .

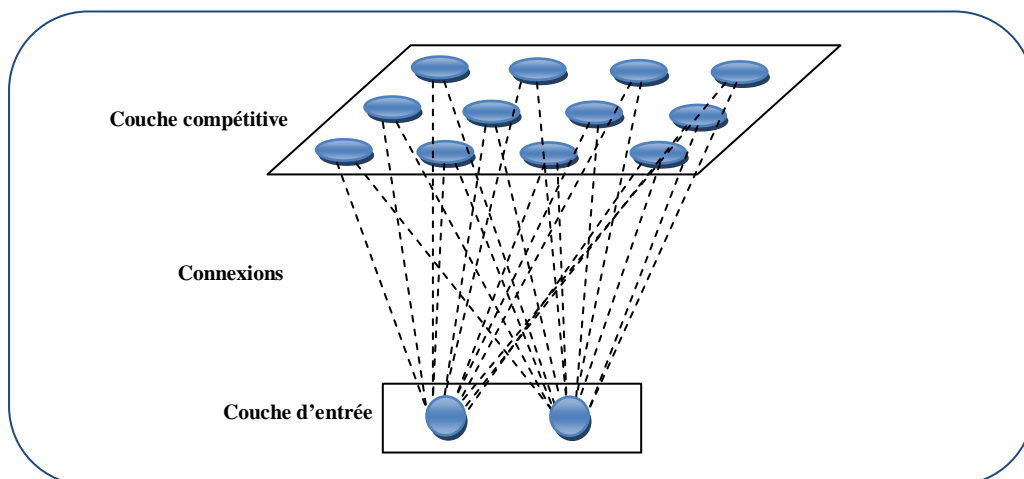


Figure 1.2. L'architecture d'une carte de Kohonen.

### 1.2.3. Les réseaux de neurones probabilistes :

Un réseau neuronal probabiliste PNN (*Probabilistic Neural Network*) est un réseau neuronal feed-forward, dérivé du réseau bayésien et d'un algorithme statistique appelé analyse discriminante de Kernel Fisher [WIKIPEDIA\_10/2016]. Il a été introduit par D.F. Specht au début des années 1990. Dans un PNN, les opérations sont organisées en un réseau Feed-forward multicouche avec quatre couches :

- ✚ Couche d'entrée.
- ✚ Couche cachée.
- ✚ Couche de modèle / Couche de somme.
- ✚ Couche de sortie.

Le réseau de neurones probabiliste est conçu selon quatre couches de neurones dont seuls les neurones de deux couches adjacentes sont interconnectés. L'information transite dans un seul sens, d'une couche  $n$  à une couche  $n+1$  et chaque neurone d'une couche est dédié à la même tâche. La première couche associe à chaque neurone une variable de l'observation nouvelle d'où ses informations sont distribuées à tous les neurones de la seconde couche. Dans cette dernière, il existe un neurone par observation apprise où chaque neurone calcule une distance euclidienne entre l'observation nouvelle et l'observation apprise, pondérée par un paramètre de lissage. Ce paramètre permet de contrôler la finesse de généralisation de la méthode. Il évolue sur l'ensemble des réels positifs non nuls et tend vers 0 lorsque le nombre d'observations d'apprentissage tend vers l'infini, en appliquant sur cette distance la fonction d'activation exponentielle. Ensuite, ces informations sont transférées à un neurone spécifique de la troisième couche, en additionnant ces informations pour chaque groupe d'observations. Une quatrième et dernière couche de neurones reçoit toutes les informations spécifiques à chaque groupe et opère la prédiction de la classe [SPECHT\_1990].

### 1.2.4. Les réseaux de neurones impulsioneels :

La compréhension actuelle des neurones biologiques est qu'ils communiquent à travers des impulsions et emploient le moment relatif des impulsions pour transmettre des informations et effectuer des calculs. Les réseaux de neurones impulsioneels SNNs (*Spiking Neural Networks*), qui sont la troisième génération de réseaux de neurones artificiels, imitent de véritables neurones biologiques du cerveau [MAASS\_1997]. Les SNNs sont des systèmes de calcul complexes (Figure 1.3), non linéaires, qui sont plus puissants sur le plan informatique que les réseaux de neurones traditionnels.

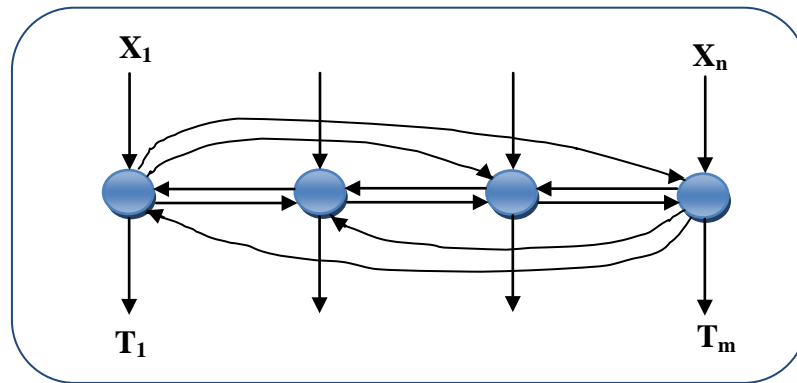


Figure 1.3. Le réseau neuronal temporel.

### 1.2.5. Les classifieurs hybrides :

Dans les années 80, un nouveau concept apparu consiste à hybrider les classifieurs pour bénéficier de leur éventuelle complémentarité. L'objectif principal de cette hybridation est l'amélioration de la précision de reconnaissance.

Parmi les classifieurs hybrides présentés dans les travaux cités en littérature, nous citons :

- ✚ Les systèmes Neuro-Flous : ils permettent de combiner les avantages de deux techniques complémentaires (les systèmes flous et les réseaux de neurones). Les systèmes flous fournissent une bonne représentation des connaissances, et l'intégration des réseaux de neurones au sein de ces systèmes améliore les performances grâce à la capacité d'apprentissage.
- ✚ Le système d'inférence floue basé sur le réseau adaptatif ANFIS (*An adaptive neuro-fuzzy inference system*) : est une sorte de réseau de neurones artificiels qui est fondée sur le système d'inférence floue de Takagi-Sugeno. Il a été développé au début des années 1990 [JANG\_1991].
- ✚ Le réseau de neurones d'ondelettes WNN : il a été introduit par Zhang et Benveniste en 1992 [ZHANG\_1992] comme une alternative aux réseaux de neurones non bouclés (feed-forward) pour l'approximation arbitraire des fonctions non linéaires. Les fonctions d'activation des neurones des couches cachées du réseau WNN sont de type ondelette.

### 1.3. La classification neuronale des données médicales :

La littérature présente une infinité de travaux sur la classification neuronale des données médicales. Notre but n'est pas de donner un bilan complet, mais simplement de mettre en lumière les principales contributions dont certaines ont été sources de notre réflexion. Nous allons classer ces travaux en deux approches ; la première consiste à programmer le classifieur neuronal dans un processeur (implantation *Software*), et la seconde vise à implémenter le classifieur neuronal matériellement dans des circuits reconfigurables du type FPGA (implantation *Hardware*).

Cette étude a été approfondie sur plusieurs cas de pathologies (Annexe A) et élargit aux différents systèmes d'aide au diagnostic, qui utilisent les réseaux neuronaux. Nous avons choisi de présenter quelques travaux réalisés pour chaque pathologie en utilisant les deux approches d'implémentation.

### 1.3.1. Les travaux réalisés sur les arythmies cardiaques :

Pour des diagnostics efficaces des pathologies cardiovasculaires, l'étude du signal ECG doit s'effectuer pendant plusieurs heures car les symptômes des pathologies peuvent se produire à tout instant, donc l'analyse et l'interprétation assistée par ordinateur des signaux ECG devient indispensable afin d'aider les médecins cardiologues à faire le diagnostic, et pour assurer une surveillance continue des patients.

Plusieurs travaux concernant la classification neuronale des signaux ECG par **l'approche Software** ont été proposés dans la littérature ces dernières années, parmi eux :

Le travail d'**Ebrahimzadeh** et **Khazae**, qui ont utilisé un réseau de neurones (MLP) avec un nombre différent de couches, en appliquant neuf algorithmes d'apprentissage. Les performances des réseaux en termes de vitesse de convergence et de précision de classification ont été validées sur sept enregistrements de la base de données MIT-BIH, le taux de classification obtenu est de 95.4% [**EBRAHIMZADEH\_2010**].

**Patra** et **al**, ont intégré dans la phase de pré-classement l'analyse en composantes principales (ACP) et le regroupement flou de c-means (FCM) afin de réduire la dimension des données au classifieur neuronal de rétro-propagation (BPNN). Ils ont appliqué une étude comparative entre quatre structures réalisées FCM-NN, PCA-NN, FCM-PCA-NN et FCM-PCA-NN. La méthode FCM-PCA-NN a donné les meilleurs résultats [**PATRA\_2010**].

**Asyiqin** et **al**, ont utilisé la transformée en ondelette discrète optimale (DWT) et un classifieur neuronal pour classifier les battements ESV et les battements normaux. Les auteurs ont montré que les ondelettes «haar», «db3 » et « sym3 » produisent des résultats prometteurs en testant 400 battements [**ASYIQIN\_2010**].

**Abibullaev** et **Seo**, ont présenté une nouvelle méthode pour la détection et la classification des complexes QRS dans les signaux ECG, ils ont utilisé la transformée en ondelettes continues (CWT) et le réseau neuronal qui a été entraîné par l'algorithme de rétro-propagation. La classification des complexes QRS normaux et anormaux était obtenue avec une précision moyenne de 97.2% [**ABIBULLAEV\_2011**]. Un autre classifieur perceptron multicouche entraîné par l'algorithme de rétro-propagation a été proposé par **Jadhav** et **al**, pour différencier entre les cas normaux et les arythmies qui apparaissent sur plusieurs signaux ECG. Les résultats expérimentaux ont donné un taux de classification de 86.67% [**JADHAV\_2012**].

Les chercheurs **Gothwal** et **al**, ont combiné la transformée de Fourier (FFT) et un réseau de neurones avec un algorithme de rétro-propagation de Levenberg Marquardt pour analyser le signal ECG. Le meilleur résultat de classification obtenu était de 98.48% [**GOTHWAL\_2011**].

D'autre part, **Özbay et al**, ont présenté un nouveau système de diagnostic automatisé, ils ont combiné la transformée en ondelettes (WT) et l'algorithme de regroupement flou de c-means du type 2 (T2FCM) pour améliorer les performances du classifieur neuronal. En utilisant ce système, ils ont obtenu un taux de classification de 99% [**ÖZBAY\_2011**].

Un modèle de classification basé sur la transformée en ondelettes continue (CWT), l'analyse en composante principale (ACP) et un réseau de neurones multicouche (MLP) a été implémenté par **Ghorbanian et al**, afin de classifier six classes de l'ECG en utilisant seulement 600 battements à partir de différents enregistrements de la base de données MIT-BIH, ils ont obtenu un taux de classification de 99.17% [**GHORBANIAN\_2011**].

**Sekkal et al**, ont proposé un classifieur neuronal qui utilise l'algorithme génétique pour la détermination des connexions optimales entre les neurones, afin d'avoir une meilleure reconnaissance des extrasystoles ventriculaire (ESV). La base de données d'arythmie MIT-BIH a été employée pour valider l'approche [**SEKKAL\_2011**]. Un autre travail de **Sekkal et Chikh**, où ils ont utilisé un classifieur neuro-génétique en employant une méthode basée sur un choix non aléatoire de la population initiale par l'apprentissage individuel, puis ils ont appliqué un apprentissage global. Cette méthode a montré une amélioration considérable sur la classification des arythmies cardiaques [**SEKKAL\_2012**].

**Rai et al**, ont proposé une technique pour classer les données du signal ECG en deux classes (classe normale et anormale) en appliquant trois classifieurs neuronaux différents : le réseau de propagation arrière (BPN), le réseau d'alimentation directe (FFN) et le perceptron multicouche (MLP). La performance du système a été obtenue avec une précision de 100% en utilisant le MLP [**RAI\_2013**].

**Wang et al**, ont réalisé une méthode de réduction des caractéristiques en combinant l'analyse en composantes principales (ACP) avec l'analyse discriminante linéaire (LDA), puis ils ont utilisé un classifieur de réseau neuronal probabiliste (PNN) pour reconnaître huit types différents d'arythmies à partir des battements ECG. Les résultats expérimentaux ont été validés avec une précision de classification satisfaisante de 99.71% [**WANG\_2013**].

Dans le travail de **Benchaib et Chikh**, ils ont appliqué l'algorithme de métaplasticité artificielle sur un perceptron multicouche AMMLP (*Artificial Metaplasticity Multi layer Perceptron*) pour classer différentes arythmies cardiaques. L'algorithme proposé est inspiré de la propriété biologique de métaplasticité des neurones et de la théorie d'information de Shannon [**BENCHAIB\_2014**].

Une nouvelle méthode pour la détection des arythmies cardiaques a été proposée par **Boussaa et al**, où ils ont classifié les images des signaux ECG normaux et anormaux en utilisant des algorithmes puissants de traitement d'images et d'intelligence artificielle : les caractéristiques pseudo-Haar (Haar-like) qui sont basées sur le concept de l'image intégrale et le classifieur neuronal de type perceptron multicouche. Les résultats expérimentaux ont montré que le système combine les avantages respectifs du descripteur de type haar et des réseaux de neurones artificiels en termes de robustesse et de vitesse [**BOUSSAA\_2015**].

D'autres travaux ont été publiés ces dernières années basés sur **l'implémentation matérielle** des classifieurs neuronaux pour la détection des arythmies cardiaques comme le travail de **Armato et al**, qui ont implémenté la carte d'auto-organisante de Kohonen (SOM) sur un FPGA du type Virtex-4 FX12 de Xilinx pour la reconnaissance des arythmies cardiaques entre classe normale et pathologique en temps réel [**ARMATO\_2009**].

**Jewajinda et Chongstitvatana**, ont proposé un classifieur neuronal basé sur un algorithme génétique parallèle du type cellulaire (EC-CGA) approprié pour une réalisation matérielle, l'apprentissage est effectué en ligne. Le classifieur est mis en œuvre sur un circuit reconfigurable du type FPGA pour une classification portable des signaux ECG [**JEWAJINDA\_2010**]. Une autre application de **Jewajinda et Chongstitvatana**, où ils ont proposé un nouvel algorithme appelé algorithme génétique Compact Cellulaire (C-CGA) pour l'apprentissage en ligne du réseau neuronal pour une implémentation matérielle. Cet algorithme est une structure de filtre numérique adaptative, il effectue également une sélection de fonctions adaptatives en temps réel. Le réseau adaptatif proposé est implémenté dans un circuit reconfigurable (FPGA) pour la classification des signaux ECG afin de faire une surveillance à long terme des patients [**JEWAJINDA\_2013**].

Ozdemir et Danisman [**OZDEMIR\_2011**], ont présenté une architecture totalement parallèle d'un perceptron multicouche implémentée sur un circuit FPGA du type Cyclone III EP3C120F780C7 d'Altera, en utilisant une représentation de données à virgule flottante. Les caractéristiques du signal ECG sont réduites par l'analyse en composantes principales (ACP), ils ont construit une base de 4 enregistrements (205, 208, 210 et 213) de la base de données MIT-BIH dont le 2/3 était utilisé pour l'apprentissage du classifieur et le reste pour le test. Le classifieur est capable de classer 3 battements d'ECG (N, V et F) avec un taux de classification de 97.66%.

Dans le papier de Jatmiko et al [**JATMIKO\_2011**], ils ont utilisé la transformation par ondelettes discrètes (DWT) pour caractériser les battements ECG, et un classifieur neuronal basé sur l'algorithme de rétro-propagation afin de classer ces battements en classe normale et anormale. Le système a été implémenté sur un FPGA du type Spartan-3ADSP XC3SD3400A de Xilinx en réalisant un taux de classification qui varie de 90% à 100%. Le même modèle a été proposé par Egila et al [**EGILA\_2016**], en ajoutant à son entrée un filtre (FIR) pour filtrer le bruit du signal ECG, ils ont utilisé une base d'apprentissage de 48 enregistrements d'ECG normaux et 42 enregistrements anormaux. Pour la base de test, 24 enregistrements normaux et 21 anormaux. Leur classifieur a été réalisé sur un FPGA en utilisant la carte XILINX Spartan 3AN-XC3S700AN. Les performances obtenues ont été présentées avec un taux de 95.23% pour l'identification des battements anormaux, et un taux de classification globale de 97.8%.

Un système de détection automatique des maladies cardiaques a été mis en œuvre par **Travieso-González et al**, sur le circuit reconfigurable FPGA. Le système était capable de traiter, d'analyser et de classer les pathologies cardiaques en temps réel à partir de l'électrocardiogramme (ECG). Le réseau neuronal utilisé était basé sur un algorithme



combinant les techniques de la transformée d'ondelettes numériques (DWT) et de l'analyse en composantes principales (ACP) [GONZALEZ\_2013].

### 1.3.2. Les travaux réalisés sur les maladies de cancer :

Dans notre étude, nous nous sommes limité seulement aux deux types de cancer : le cancer du sein (*Breast Cancer*) et le cancer du cerveau (*Brain Cancer*).

La classification et la détection des tumeurs dans différentes images médicales par **l'implémentation software** sont motivées par la nécessité d'une grande précision dans le diagnostic de la maladie. Pour cela, de nombreuses recherches sur les techniques intelligentes de la classification des anomalies dans les images de mammographie et d'IRM ont été présentées ces dernières années :

**Nithya** et **Santhi**, ont proposé une méthode de diagnostic de cancer du sein en utilisant les techniques de traitement d'images qui se base sur la matrice de Co-Occurrence de niveau de gris (GLCM). Les images de mammographie ont été classifiées comme normales ou cancéreuses par un classifieur neuronal. La base de données DDSM (*Digital Database for Screening Mammography*) a été utilisée afin d'évaluer le système développé, le taux d'exactitude maximal pour la classification du cancer du sein est de 96% [NITHYA\_2011].

Les chercheurs **Marcano** et **al**, ont présenté une nouvelle amélioration de l'apprentissage des réseaux de neurones pour la classification des modèles. L'algorithme proposé est inspiré de la propriété de la métaplasticité biologique des neurones et de la théorie de Shannon, il a été appliqué sur un perceptron multicouche et la performance de ce classifieur a été évaluée sur la base de données WBCD (*Wisconsin Breast Cancer Database*), il ont obtenu une précision de 99.26% [MARCANO\_2011].

Une nouvelle approche de classification pour la détection des anomalies mammaires dans les mammographies numériques a été présentée par **Dheeba** et **al**, en utilisant le réseau neuronal d'ondelettes optimisé par particules (PSOWNN). L'algorithme de détection des anomalies proposé repose sur l'extraction des mesures de l'énergie des textures des mammographies et la classification des régions suspectes en appliquant un classifieur de motifs. La méthode est appliquée sur une base de données cliniques réelle de 216 images recueillies à partir de centres de dépistage de mammographie. Ils ont obtenu respectivement une sensibilité de 94.167% et une spécificité de 92.105% [DHEEBA\_2014].

D'autres chercheurs comme **Hassanien** et **al**, ont utilisé des images d'IRM mammaires afin de classifier la nature du cancer. Ils ont proposé une approche hybride qui combine les avantages des ensembles flous du type II pour améliorer le contraste des images d'entrée, l'algorithme de colonie de fourmis pour segmenter les objets cibles et un classifieur perceptron multicouche MLP afin de déterminer le type de cancer (bénin ou malin) [HASSANIEN\_2014].

**Ourdighi** et **Benyettou** ont proposé un réseau de neurones impulsionsnels (SNN) à une seule couche, basé sur la généralisation SRM (*Spike Response Model*) avec plusieurs retards

par connexion. L'approche a été testée sur la base de données WBCD, ils ont obtenu un taux de classification de 99.26% [OURDIGHI\_2016].

Récemment, **Singh et al**, ont présenté une nouvelle technique de classification en combinant la technique d'apprentissage non supervisée (le regroupement flou de c-means (FCM)) et la technique d'apprentissage supervisée (le réseau de neurones artificiels de rétro-propagation) pour classifier les tumeurs bénignes et malignes dans les images échographiques. Les expériences ont été réalisées sur 178 images d'échographie mammaire contenant 88 cas bénins et 90 malins. Ils ont atteint une précision de classification de 95.862% [SINGH\_2016].

**Dipali et al**, ont utilisé différentes techniques de traitement d'images pour la détection de la tumeur cérébrale dans les images IRM. L'extraction des caractéristiques de texture de la tumeur ciblée a été obtenue en utilisant la matrice de Co-occurrence de niveau de gris (GLCM), et un classifieur Neuro-floue a été développé pour identifier les différents types de cancers du cerveau. Le système a été évalué par les images d'IRM des patients atteints de cancer du cerveau, ces images ont été obtenues du service de radiologie de TMH (*Tata Memorial Hospital*) en Inde. Le système a été jugé efficace pour la classification de ces échantillons [DIPALI\_2010].

Un réseau neuronal probabiliste (PNN) a été utilisé par **Othman et Ariffanan** avec des techniques de traitement d'images pour mettre en place une classification automatisée des tumeurs cérébrales. Le système a été testé sur des images IRM, le temps de réponse était rapide et ils ont obtenu une bonne précision [OTHMAN\_2011].

**Hemanth et al**, ont appliqué des systèmes d'inférences neuro-floue adaptative (ANFIS) pour la classification des tumeurs cérébrales de l'IRM. Ils ont utilisé des images anormales de tumeurs cérébrales provenant de quatre classes. Un ensemble complet de caractéristiques et des règles floues sont sélectionnés pour identifier une image anormale au type de tumeur correspondant. Les résultats étaient prometteurs en termes de précision de la classification [HEMANTH\_2011].

Dans Le papier présenté par **Sridhar et Krishna**, les auteurs ont utilisé la transformation de Cosinus discrète pour la réduction de dimensionnalité et l'extraction de caractéristiques des images IRM de tumeurs cérébrales, et le réseau neuronal probabiliste (PNN) pour la classification de ces images. L'évaluation a été effectuée sur une base de 20 images de tumeurs cérébrales. La méthode proposée a donné un taux de reconnaissance rapide et meilleur [SRIDHAR\_2013].

**Aniket et al**, ont utilisé différentes techniques de traitement d'images telles que l'égalisation d'histogramme, segmentation d'image et les opérations morphologiques pour améliorer les images IRM. Ils ont appliqué un classifieur neuronal pour la reconnaissance des tumeurs cérébrales dans les images IRM des patients. Les résultats de la classification étaient en accord avec l'avis des médecins [ANIKET\_2013].

**Amsaveni et Singh**, ont proposé une technique automatique permettant de définir la région tumorale dans les images IRM du cerveau. L'algorithme intègre des étapes pour le prétraitement, l'extraction de caractéristiques et la classification. L'extraction des caractéristiques de texture dans la tumeur détectée a été réalisée en utilisant le filtre de Gabor, et ces caractéristiques sont utilisées pour classifier la tumeur cérébrale, employant le classifieur neuronal artificiel. Le système améliore significativement la précision de classification et la détection de tumeurs cérébrales [AMSAVENI\_2014].

Le système proposé par **Aqhsa et Narayanan** comporte plusieurs étapes : la segmentation des images IRM provenant des tissus cérébraux normaux et anormaux, l'extraction de vecteurs de caractéristiques et la classification des modèles par un perceptron multicouche. La méthode de détection des tumeurs cérébrales présentée permet d'augmenter la précision et le rendement et de diminuer le temps de diagnostic [AQHSA\_2014].

Dans le contexte de **l'implémentation matérielle**, nous constatons différents travaux sur la classification du cancer du sein ainsi que les tumeurs cérébrales par les réseaux de neurones, comme le travail présenté par **Docusse et al**, ils ont proposé un système de diagnostic assisté par ordinateur qui classe automatiquement cinq types de micros-calcifications détectées sur les mammographies numériques. Le système développé utilise la combinaison d'ondelettes et les réseaux neuronaux artificiels (ANNs), qui a été implémenté par la suite sur un kit de développement Altera DE2-115. Ce dernier contient un circuit reconfigurable (FPGA) permettant au système d'être plus petit, moins cher et plus efficace énergétiquement. Les résultats ont montré que le système a classé correctement 96.67% des échantillons d'essai. Ce système peut être utilisé comme un second test pour les radiologues lors du diagnostics précoces de cancer du sein [DOCUSSE\_2013].

**Selvathi et al**, ont implémenté un classifieur neuronal de type perceptron multicouche qui a utilisé l'algorithme de rétro-propagation sur un FPGA Virtex5 de Xilinx. La conception matérielle a été réalisée à l'aide du langage Verilog et les performances du classifieur ont été évaluées en utilisant la base de données du cancer du sein WBCD (*Wisconsin Breast Cancer Database*). Ils ont obtenu un taux de classification de 90.83% [SELVATHI\_2016].

**Menaka et Rohini**, ils ont traité la détection de l'AVC ischémique, qui se produit à cause d'un blocage dans les artères du cerveau humain. Les images IRM obtenues sont prétraitées et améliorées, puis les caractéristiques des images sont calculées au moyen de la matrice de Co-occurrence de niveau de gris (GLCM). Ils ont réalisé un réseau de neurones pour classer les images IRM en normales et anormales en observant une efficacité de 90%. Le classifieur neuronal a été implémenté sur un circuit reconfigurable FPGA du type Spartan 3E en utilisant l'environnement Design 14.3 de la société Xilinx [MENAKA\_2014].

Dans le papier de **Gouri et al**, ils ont utilisé un DSP (*Digital signal processing*) qui se trouve dans le générateur du système de Xilinx XSG (*Xilinx System Generator*) pour améliorer la qualité des images du cerveau humain. Ensuite, ces images sont classées en normales et anormales en utilisant un perceptron multicouche entraîné par l'algorithme de rétro-propagation. L'architecture présentée utilise une interface graphique qui relie Simulink de Matlab et le XSG de Xilinx en offrant une implémentation matérielle appropriée. La

performance du classifieur est évaluée en terme des ressources utilisées par un FPGA du type XC5VLX50T [GOURI\_2015].

### 1.3.3. Les travaux réalisés sur le diabète :

Nous présentons quelques travaux qui ont utilisé les réseaux de neurones pour la reconnaissance du diabète, les chercheurs **Pradhan** et **Sahu** ont proposé un perceptron multicouche et l'algorithme génétique pour classer les sujets diabétiques ou non. Cette classification a été appliquée sur la base de données du diabète PIMA (*Pima Indian Diabetes Data*), ils ont obtenu une précision de 72.2% [PRADHAN\_2011].

**Dogantekin** et **al**, ont utilisé la méthode d'analyse discriminante linéaire et le classifieur ANFIS pour la reconnaissance du diabète, ils ont obtenu un taux de classification de 84.61% [DOGANTEKIN\_2010]. Par contre **Ubeyli** a obtenu un taux de 98.14% en appliquant le classifieur ANFIS et la technique de validation croisée de K-Fold sur la base de données PIMA [UBEYLI\_2010]. Ensuite, les chercheurs **Settouti** et **al**, ont utilisé la méthode de regroupement Fuzzy C-means (FCM) au système d'inférence neuro-flous adaptatif (ANFIS) afin d'augmenter l'interprétabilité du classifieur. La performance du classifieur a été évaluée en utilisant la base de données du diabète (UCI Machine Learning) [SETTOUTI\_2012].

Le travail présenté par **Belarouci**, propose une méthode de pondération basée sur l'algorithme des moindres carrés moyens LMS (*Least Mean Square*) dans le but d'affecter des poids forts aux différents échantillons de la classe minoritaire et des poids faibles aux différents échantillons des classes majoritaires afin de traiter le problème d'apprentissage des données déséquilibrées. Après cette phase d'équilibrage, ils ont utilisé plusieurs approches de classification, comme le perceptron multicouche (MLP) qui a réalisé une bonne reconnaissance des données minoritaires et des données majoritaires, la performance du classifieur a été évaluée en utilisant la base de données du diabète (PIMA) à un taux de classification de 99.24% [BELAROUCI\_2016].

**Akanksha** a présenté un système qui emploie une interface floue en cascade avec un réseau de neurone Feed-Forward afin d'obtenir une décision optimale concernant l'état physiologique et pathologique futur d'un patient. Le classifieur neuro-flou a été implémenté sur un FPGA pour une détection à temps réel de l'état critique d'un sujet diabétique. La précision du système a été confirmée en prédisant l'état diabétique d'un patient 30 jour avant l'état critique [AKANKSHA\_2016].

### 1.3.4. Les travaux réalisés sur les troubles neurologiques :

Dans cette étude, nous présentons quelques travaux qui ont abordé la reconnaissance de certaines maladies neurologiques telles que la maladie de Parkinson et d'épilepsie.

Dans le travail de **Resul Das**, différents classifieurs ont été réalisés pour distinguer efficacement les individus en bonne santé des gens atteints de la maladie de parkinson comme le réseau neuronal et l'arbre de décision. Différentes méthodes d'évaluation ont été utilisées

pour calculer la performance des classifieurs, le classifieur neuronal a donné les meilleurs résultats avec un taux de 92.9% [DAS\_2010].

Pour identifier la présence de la maladie de Parkinson, **Mehmet Can**, a présenté un système de réseau neuronal en utilisant l'algorithme de rétro-propagation et la méthode de vote majoritaire. Le classifieur a donné un taux de reconnaissance supérieur à 90% [CAN\_2013].

Les chercheurs **Olanrewaju et al**, ont utilisé un réseau de neurones feedback multicouche (MLFNN) avec l'algorithme de rétro-propagation et l'algorithme de regroupement de K-Means pour une détection précoce de la maladie de Parkinson. Les performances du classifieur ont été évaluées en utilisant la base de données prise d'Oxford (*Parkinson's Disease Detection Dataset*), ils ont obtenu une sensibilité de 83.3%, une spécificité de 63.6% et 80% pour la précision [OLANREWAJU\_2015].

**Saloni et al**, ont comparé deux classifieurs réalisés pour la détection de la maladie de Parkinson : le réseau neuronal avec différentes fonctions d'activation et le classifieur SVM (*Support Vector Machines*). Ils ont sélectionné 15 caractéristiques aux entrées des deux classifieurs parmi les 23 descripteurs [SALONI\_2015].

**Orhan et al**, ont comparé entre différentes méthodes de classification pour le diagnostic efficace des maladies de Parkinson : le système immunitaire artificiel, le réseau Feed-forward, la quantification de vecteurs d'apprentissage et le réseau neuronal probabiliste. Ils ont utilisé 22 descripteurs et la technique de validation croisée (10 fois) afin d'estimer les performances des classifieurs. Les taux de la classification respectifs sont 95.6%, 95.4%, 91.4% et 96.5%, le meilleur résultat de classification était obtenu par le réseau neuronal probabiliste [ORHAN\_2016].

**Zainuddin et al**, ont étudié la faisabilité et l'efficacité des réseaux de neurones d'ondelettes (WNN) dans la détection des crises épileptiques. Les signaux d'électroencéphalographie (EEG) ont d'abord été prétraités par les transformées d'ondelettes discrètes (DWT). Ils ont réalisé trois classifieurs dont les fonctions d'activation des neurones cachés sont respectivement l'ondelette gaussienne, le chapeau mexicain et l'ondelette de Morlet. Une validation croisée a été réalisée et l'évaluation de la performance a révélé que les classifieurs proposés ont obtenu une précision de classification globale élevée. La meilleure combinaison était le WNN qui utilisait l'ondelette de Morlet comme fonction d'activation [ZAINUDDIN\_2012].

**Omerhodzic et al**, ont proposé un classifieur neuronal pour la reconnaissance des signaux EEG (sujets en bonne santé, patients atteints d'épilepsie et patients souffrant de syndrome épileptique pendant la crise). Ils ont combiné avec le classifieur neuronal la transformée d'ondelettes discrète (DWT) avec l'analyse multi-résolution (MRA) pour décomposer le signal EEG, ils ont utilisé aussi le théorème de Parseval pour extraire la répartition en pourcentage des caractéristiques énergétiques. La performance de l'algorithme proposé a été évaluée en utilisant au total 300 signaux EEG et les résultats ont montré que le classifieur

proposé a la capacité de reconnaître et de classifier efficacement les signaux EEG [OMERHODZIC\_2013].

**Ouelli et al**, ont présenté une nouvelle méthode de détection des crises épileptiques, basée sur la modélisation autorégressive des signaux ischémiques et sans crises, un perceptron multicouche qui a été réalisé par l'algorithme Levenberg-Marquardt (LM). L'approche proposée a donné une précision de classification de 94.67% [OUELLI\_2015].

Des modèles de réseaux de neurones artificiels ont été appliqués par **Dhande et Gulhane** pour la classification des signaux EEG épileptiques, l'approche de conception consiste à augmenter le nombre de neurones cachés et de trouver la valeur optimale. L'apprentissage a été effectué par l'algorithme de rétro-propagation et les performances du classifieur ont été évaluées à l'aide d'une base de données collectée au niveau des hôpitaux d'université de Bonn en Allemagne [EPILEPT\_2001]. La méthode proposée fournit une bonne précision de classification et une caractéristique de généralisation très correcte [DHANDE\_2015].

Autres travaux concernant **l'implémentation matérielle** des classifieurs neuronaux pour la détection des crises épileptiques ont été présentés ces dernières années, nous citons le document de **Saleheen et al**, ils ont proposé une conception matérielle d'un classifieur neuronal. Ils ont utilisé la fonction statistique de variance pour extraire les caractéristiques. Le classifieur a été implémenté dans un FPGA, ils ont obtenu une précision de détection de 99.18% avec une utilisation matérielle acceptable (environ 44% des ressources FPGA) [SALEHEEN\_2010].

**Harikumar et al**, ont mis en œuvre des réseaux de neurones à ondelettes (WNN) avec la capacité d'apprentissage sur un circuit reconfigurable pour la détection d'épilepsie. Les signaux d'électroencéphalographie (EEG) ont été d'abord prétraités en utilisant les transformées en ondelettes discrètes (DWT). Trois fonctions d'activations différentes ont été utilisées dans les neurones cachés des réseaux (WNN) : la fonction Gaussienne, le chapeau mexicain et les ondelettes de Morlet. Ils ont choisi pour ces réseaux l'apprentissage par l'algorithme de PSO (*Particle Swarm Optimization*). Les réseaux ont été implémentés sur un circuit reconfigurable FPGA pour prendre l'avantage du parallélisme. La meilleure performance de reconnaissance d'épilepsie a été détectée dans le réseau de neurones qui utilisait l'ondelette de Morlet comme fonction d'activation [HARIKUMAR\_2013].

Récemment, **Kueh et Kazmierski**, ont montré l'intérêt de rendre les éléments du réseau neuronal en parallèle à travers du matériel en utilisant le traitement « bit-serial ». La mise en œuvre de ces éléments de traitement neuronal série sur un circuit FPGA du type Cyclone IV d'Altera dénombre 16 éléments logiques pour les multiplicateurs, 30 éléments logiques pour les générateurs de fonction (*Look Up Table*) et 55 éléments logiques pour le contrôleur. L'architecture matérielle du réseau de neurones peut être mise en œuvre dans un équipement portable afin de répondre aux besoins des patients épileptiques dans leurs activités [KUEH\_2016].

## 1.4. Les avantages et les limites des approches d'implémentation :

Dans la section précédente, de nombreux travaux sur la classification neuronale ont été présentés et classés selon deux scénarios, logicielle et matérielle, afin d'aider l'expert dans son diagnostic médical. Ces approches ont été proposées selon le mode de traitement de l'information, soit d'une façon séquentielle ou parallèle.

### 1.4.1. Approche Software (logicielle) :

La plupart des classifieurs neuronaux sont développés sur des ordinateurs classiques, des DSP (*Digital Signal Processor*) et des microcontrôleurs, à l'aide de différents langages (Matlab, Langage C, etc.) en vue d'une utilisation logicielle. L'architecture interne de ces circuits est composée d'opérateurs arithmétiques et logiques, registres, modules dédiés au contrôle et synchronisation des opérations, et éventuellement de mémoires internes afin de stocker les données et les résultats. Ce type de conception est totalement axé sur les performances de classification, et il ne possède pas de réelle limitation sur l'utilisation mémoire ou encore sur le coût calculatoire des algorithmes.

Dans ces circuits, le programme de contrôle détermine les opérations à exécuter, offrant une grande flexibilité d'application. Le fonctionnement séquentiel est un inconvénient majeur pour les performances de ces circuits lorsque nous envisageons une implémentation d'algorithmes strictement parallèles. En effet, le processeur est constitué de telle façon qu'il exécute les différentes instructions de façon séquentielle.

### 1.4.2. Approche Hardware (matérielle) :

Les systèmes embarqués dédiés au domaine médical sont réalisés pour répondre à la demande croissante des systèmes d'aide au diagnostic. Ces systèmes embarqués sont le plus souvent des composants dotés d'architectures parallèles (ASIC, FPGA) qui peuvent être utilisés pour accélérer les calculs de traitement des données médicales.

La mise en œuvre des classifieurs neuronaux sur des circuits du type ASIC (*Application Specific Integrated Circuit*) permet généralement l'obtention de meilleures performances en termes de temps de calcul, consommation et ressources. Mais les inconvénients majeurs d'une conception optimisée résident dans le fait du passage obligatoire chez le fondeur, ce qui implique des frais de développement élevés du circuit, et il s'avère nécessaire d'être rigoureux lors de la phase de développement, de telle sorte que le circuit prototype fonctionne dès les premiers essais et sans aucun changement dans le cahier des charges. Pour cela, l'approche matérielle à base d'ASIC a été écartée de notre étude.

Les circuits FPGA apparaissent comme étant une excellente alternative pour les applications de haute performance ayant des volumes de productions faibles ou moyennes. La technologie des circuits FPGA a connu une évolution rapide depuis plusieurs années, dans différents domaines : militaire, médical, télécommunication... Ils sont du type reconfigurable, composés de nombreuses cellules logiques élémentaires, dont elles peuvent être assemblées librement afin d'implémenter une fonction. L'augmentation des fréquences

d'horloge et la possibilité d'exploiter massivement le parallélisme, à permettre aux FPGA d'avoir une très grande flexibilité, les FPGA sont en mesure de délivrer des performances proches de celles obtenues par les circuits ASIC.

En médecine, l'aide au diagnostic consiste en l'extraction de connaissances dans de grands volumes de données pour en extraire des informations pertinentes. Ainsi, de très nombreuses applications d'aide à la décision ont été développées dans ce domaine, ces applications sont destinées à soutenir l'expert dans sa prise de décision. Cela implique l'utilisation de divers outils d'aide au diagnostic, en se basant sur l'une des deux approches présentées, logicielle ou matérielle. Ces approches sont comparées dans le tableau suivant :

**Tableau 1.1. Comparaison entre les deux approches d'implémentation.**

Les points de comparaison	Approche Software		Approche Hardware	
	DSP et Microcontrôleurs	Ordinateur	FPGA	ASIC
Mode de traitement	Séquentiel		Parallèle	
Temps de calcul	Rapide (temps réel. (ns))	Moyen (ms)	Très Rapide (temps réel. (ns))	Très Rapide (temps réel. (ns))
Flexibilité	Très flexible	Très flexible	flexible	Rigide
Programmabilité	Programmable	Programmable	Reprogrammable	Fixe
Embarquabilité	Facile à embarquer	Ne peut pas être embarqué	Facile à embarquer	Facile à embarquer
Le système utilisé	Système d'exploitation temps réel (RTOS)	Système d'exploitation classique (Défaillance possible)	Aucun (Système câblé)	Aucun (Système câblé)

L'approche matérielle à base de FPGA a été choisie pour des raisons évidentes de flexibilité, de reprogrammabilité et de rapidité.... Le parallélisme des réseaux de neurones peut facilement être exploité sur ce type de circuits. En effet, les FPGA constituent donc un candidat idéal pour la classification neuronale des données médicales.

## 1.5. Conclusion :

Nous avons remarqué à partir des travaux de la littérature que différentes techniques de classification des données médicales peuvent aider les médecins ou les radiologues à établir un deuxième diagnostic pour confirmer la pathologie.

Ces techniques ont été présentées sous deux scénarios, le premier était de programmer le classifieur neuronal des pathologies sous différents environnements, et le second était l'implémentation matérielle du classifieur neuronal dans des circuits reconfigurables du type FPGA. En choisissant cette dernière approche, la prise de décision sur n'importe quelle



maladie se fait en temps réel et sans difficulté grâce à une implémentation basée sur le parallélisme du circuit.

Le chapitre suivant présentera une description détaillée de la technologie et l'environnement de développement des circuits FPGA.



*Chapitre 2*

*Technologie et Environnement  
de Développement des FPGA.*

## 2.1. Introduction :

Lors des applications pratiques, les fabricants des circuits intégrés visent toujours à fournir des circuits présentant des densités d'intégration plus élevée, pour des vitesses de fonctionnement de plus en plus grandes. Ils ont commencé par des circuits SSI (*Small Scale Integration*) en technologie bipolaire, puis les circuits MSI (*Medium Scale Integration*) en utilisant les transistors MOS. La facilité d'intégration de ce dernier a permis la réalisation des circuits LSI (*Large Scale Integration*) puis les VLSI (*Very Large Scale Integration*). Ces deux dernières générations ont vu l'avènement des microprocesseurs et microcontrôleurs.

Les microprocesseurs et les microcontrôleurs ont bouleversé l'électronique numérique par la possibilité de réaliser n'importe quelle fonction par programmation en traitant l'information de manière séquentielle mais malheureusement la contrainte du temps est toujours posée. Pour cela, au début des années quatre-vingt, les circuits programmables PLD faisaient leurs apparitions à côté des circuits intégrés spécifiques ASIC afin de réaliser des fonctions complexes en menant l'information d'une manière parallèle. Ainsi, nous distinguons les circuits numériques de haute densité d'intégration selon le mode de traitement de l'information (Figure 2.1).

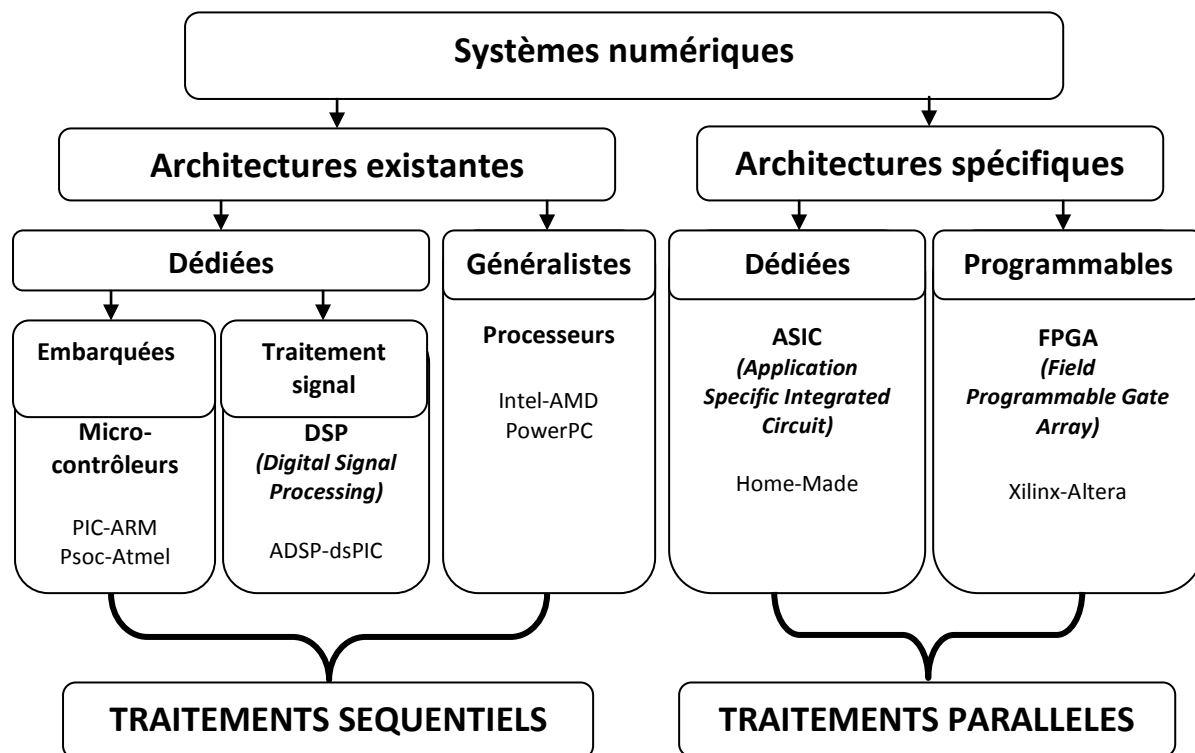


Figure 2.1. Classification des systèmes numériques de haute densité.

Nous présentons dans ce chapitre les circuits reconfigurables du type FPGA, la méthode de conception ainsi que les outils de développements.

## 2.2. Les réseaux logiques reconfigurables FPGA :

Au début des années 70, l'apparition de la connexion logique programmable a donné naissance à une industrie nouvelle qui n'a pas cessé de croître : c'est l'industrie des différents circuits logiques programmables car le premier critère de choix de ces derniers est la technologie utilisée pour matérialiser les interconnexions [DUTRIEUX\_1997].

Les technologies d'interconnexions déterminent les aspects électriques de la programmation :

- ✓ La possibilité de modifier la fonction programmée.
- ✓ Le maintien de la fonction programmée en cas de coupure d'alimentation.
- ✓ La nécessité d'utiliser un programmeur spécial.

Au milieu des années 80 les circuits du type FPGA réalisent la croisée des raisonnements génériques (PAL, CPLD) et spécifiques (ASIC) et profitent des avantages des deux technologies.

Les FPGA sont des circuits constitués de cellules interconnectées. La fonction de ces cellules ainsi que les interconnexions sont programmables ce qui donne une plus grande souplesse d'utilisation que les PLD.

Nous distinguons trois grands types de FPGA :

### 2.2.1. Les FPGA à mémoire statique :

Au début des années 80, la société XILINX a commercialisé pour la première fois un circuit FPGA sous la dénomination de LCA (*Logic Cell Array*) (Figure 2.2), elle a continué parmi d'autres sociétés à élargir sa gamme, elle propose actuellement une grande variété de produits qui utilisent la technologie de la mémoire vive [DOUILLARD\_2005].

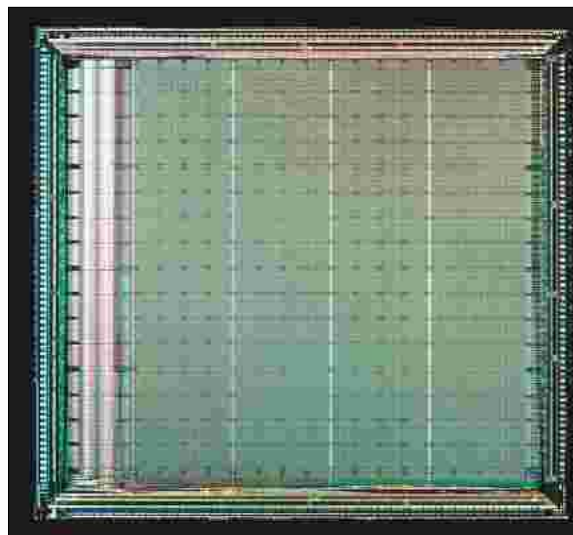
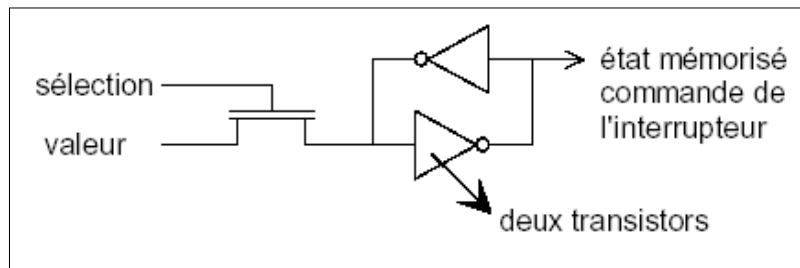


Figure 2.2. FPGA XILINX.

Dans les technologies à mémoire statique, l'état de chaque interrupteur est commandé par une cellule mémoire classique qui comporte cinq transistors : deux couples de transistors

constituent chacun un inverseur logique et l'ensemble de ces deux inverseurs bouclés entre eux réalise un bistable statique, plus un transistor de programmation, son schéma de principe est illustré par la figure 2.3.



**Figure 2.3. Cellule SRAM.**

La société XILINX est l'inventeur des connexions à SRAM [DARCHE\_2004]. L'intérêt de ces cellules est qu'elles sont programmables à volonté, mais l'information est volatile. Ces circuits permettent des reconfigurations, partielles ou totales, en nombre illimité. Il est même envisageable de créer des fonctions dont certains paramètres sont modifiables en cours de fonctionnement, des filtres adaptatifs par exemple.

### 2.2.2. Les FPGA à mémoire Flash :

Lattice Semiconductor et Actel offrent des FPGA qui enregistrent leur configuration dans une mémoire flash, un type de EEPROM (*Electrically Erasable Programmable Read-Only Memory*). Puisque la mémoire flash est une mémoire non volatile, le FPGA conserve sa configuration même lorsqu'il n'est pas alimenté.

La technologie Flash repose sur le principe de fonctionnement des transistors MOS à grille flottante, ces derniers disposent de deux grilles (Figure 2.4), dont l'une est isolée (grille d'un transistor MOS classique) et l'autre flottante (la grille de programmation). La programmation consiste à piéger les électrons dans la grille flottante en appliquant une tension de 13v entre cette dernière et la source, et une faible tension ( $\approx 5v$ ) entre le drain et la source ce qui rend le canal conducteur ; le fort champ électrique entre la grille et le canal dévie les électrons présents dans ce dernier, ils traversent la couche d'isolant de faible épaisseur ( $\approx 200\text{\AA}$ ) en se piégeant dans la grille flottante [DUTRIEUX\_1997].

- La grille flottante ne contient aucun électron => le transistor n'a pas été programmé.
- La grille flottante contient des électrons => le transistor a été programmé.

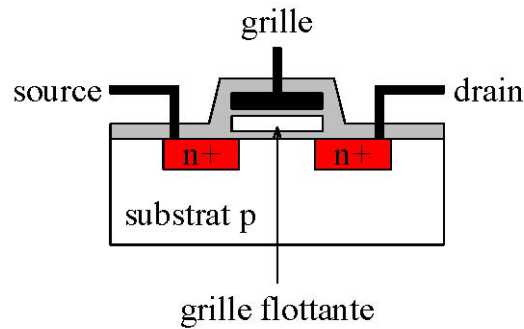


Figure 2.4. MOS à grille flottante.

### 2.2.3. Les FPGA à anti-fusibles :

Les circuits à anti-fusibles partagent, avec ceux à SRAM, le sommet de la gamme des circuits programmables en vitesse et en densité d'intégration. Il est clair que ces circuits ne sont programmables qu'une fois.

Le procédé des cellules anti-fusibles a été élaboré par la société ACTEL en 1986 sous l'appellation PLICE™ (*Programmable Low Impedance Circuit Element*) [DUTRIEUX\_1997]. La technologie repose sur un circuit ouvert (en opposition aux cellules à fusible) réalisé avec un isolant ou diélectrique (Figure 2.5), et la programmation va donc permettre d'établir la connexion.

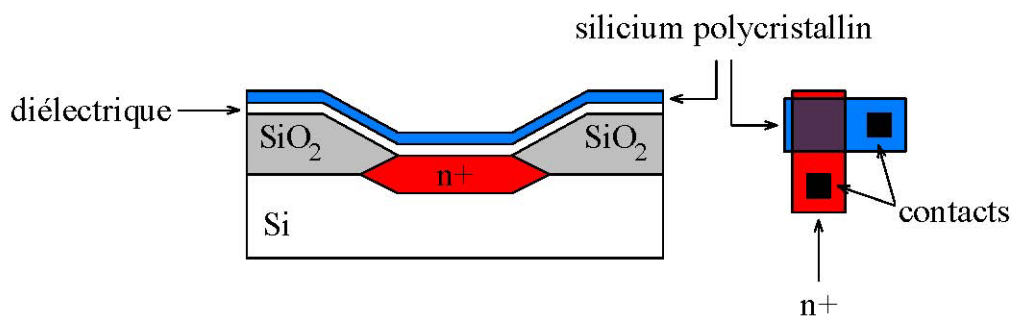


Figure 2.5. La technique des anti-fusibles.

La technique de l'anti-fusible consiste à isoler deux lignes de connexions par une fine couche d'oxyde (isolant). Il n'y a donc pas de liaison électrique entre ces lignes de connexions. Par contre si une impulsion de haute tension (une vingtaine de volts) est appliquée à cet anti-fusible, la couche d'oxyde est trouée et les deux lignes sont reliées. La programmation est donc irréversible ce qui peut être un inconvénient. Nous trouvons les deux sociétés ACTEL et ALTERA qui ont adapté la technologie anti-fusible.

### 2.2.4. Les avantages et les inconvénients de différents types de FPGA :

Dans le tableau suivant, nous présentons les avantages et les inconvénients de chaque type de FPGA.

Tableau 2.1. Comparaison entre les trois types de FPGA.

Les types de FPGA	Avantages	Inconvénients
FPGA à SRAM	<ul style="list-style-type: none"> <li>• Configuration très rapide de qq ns.</li> <li>• Nombre de configurations illimité (idéal pour le prototypage).</li> <li>• Chargement d'une configuration partielle ou totale possible en cours d'utilisation (Run Time Configuration = reconfiguration dynamique).</li> <li>• ISP : In Situ Programmable.</li> </ul>	<ul style="list-style-type: none"> <li>• Point de configuration complexe avec de nombreux transistors.</li> <li>• Mauvaises performances électriques (R et C élevés) =&gt; consommation de puissance élevée.</li> <li>• Mémoire volatile, nécessite donc une programmation à chaque mise sous tension (mémoire de configuration FLASH ou ROM externe).</li> <li>• L'échange du bitstream est une faille en termes de sécurité (protection du design).</li> <li>• Grande sensibilité aux radiations.</li> </ul>
FPGA à Flash	<ul style="list-style-type: none"> <li>• Mémoire non volatile.</li> <li>• Temps de configuration faible (d'où le nom de "Flash") de quelques <math>\mu</math>s.</li> <li>• Temps d'effacement 1000 fois plus grand (qq ms).</li> <li>• Plusieurs cellules sont configurées simultanément.</li> </ul>	<ul style="list-style-type: none"> <li>• Nécessite une alimentation supplémentaire (la tension de programmation et d'effacement est de ~10 Volts) qui peuvent être intégrés au composant sous la forme d'un système de pompe de charge.</li> <li>• Peut-être sensible aux radiations.</li> <li>• Reconfiguration peu flexible.</li> </ul>
FPGA Anti-fusible	<ul style="list-style-type: none"> <li>• Le point de configuration est très simple.</li> <li>• La configuration est rapide.</li> <li>• Technologies peu chers.</li> <li>• Très bonnes performances électriques (R et C faibles) =&gt; faible consommation (Watt).</li> <li>• Technologie très sûre pour la propriété intellectuelle (espionnage industriel).</li> <li>• Peu sensible aux radiations.</li> </ul>	<ul style="list-style-type: none"> <li>• Programmable une seule fois.</li> <li>• Tension élevée pour la configuration.</li> </ul>

### 2.3. Les types d'architectures des FPGA :

Les FPGA sont bien distincts des autres familles de circuits programmables tout en offrant le plus haut niveau d'intégration logique, car ils sont constitués d'un ensemble d'unités fonctionnelles qui, une fois programmées, permet de faire des opérations au niveau des bits.

La disposition des unités fonctionnelles dans le FPGA permet d'avoir différentes architectures de cette dernière, classiquement nous trouvons trois topologies différentes :

### 2.3.1. Architecture de type îlots de calcul :

Dans cette architecture, le FPGA est constitué d'une matrice plane d'éléments. Ces différents éléments (que l'on détaillera par la suite) constituent les ressources logiques et des routages programmables du FPGA. Ce type d'architecture a été choisi dès le départ par XILINX [XILINX\_2000].

### 2.3.2. Architecture du type hiérarchique :

Dans ce type d'architecture, il existe plusieurs plans dans le FPGA qui ne sont pas physiques, ils correspondent au niveau de hiérarchie logique. C'est-à-dire qu'un élément d'un niveau logique peut contenir des éléments de niveau logique inférieur et Chaque niveau logique reprend la topologie d'une architecture du type d'îlots de calcul avec un routage dédié pour chaque niveau. Cette architecture a été proposée par ALTERA [ALTERA\_2002].

### 2.3.3. Architecture de type mer de portes :

Ce type de topologie a été utilisé par XILINX pour sa série 6000 [GHOZZI\_2003]; il est composé hiérarchiquement et le routage est du type logarithmique. Mais ces composants n'ont pas eu de succès, commercialement parlant, par manque d'outils de CAO capables de les exploiter correctement.

## 2.4. Les couches de FPGA :

Dans le but de proposer une modélisation de FPGA, il paraît évident à connaître leur constitution (éléments de base). Lors du travail de cette thèse, nous avons choisi le FPGA du type SRAM de XILINX. L'architecture, retenue par XILINX (Figure 2.6), se présente sous forme de deux couches :

- ✓ une couche appelée circuit configurable,
- ✓ une couche réseau mémoire SRAM.

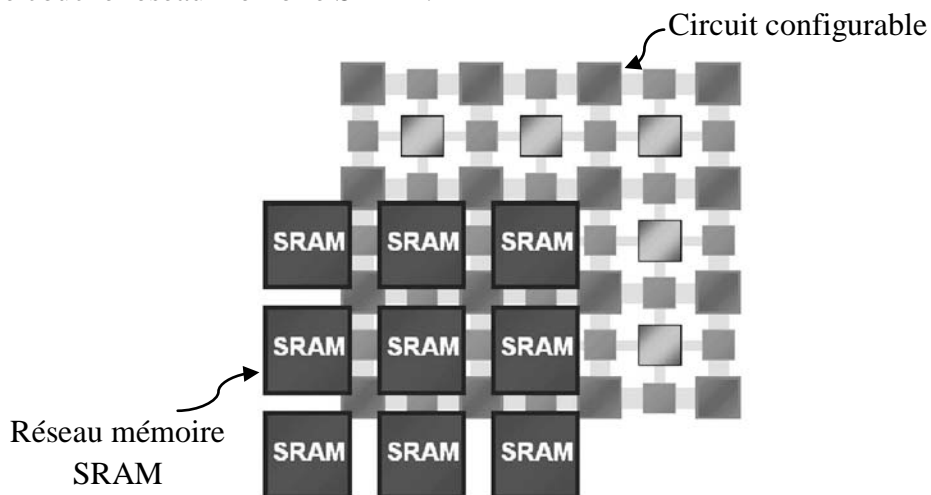


Figure 2.6. Le concept d'un FPGA du type SRAM [XILINX\_2000].



### 2.4.1. Le circuit configurable :

Le circuit configurable (Figure 2.7) est composé de trois types d'éléments programmables :

- ✓ Les blocs logiques CLB (*Configurable Logic Blocks*) qui permettent la réalisation des fonctions combinatoires et séquentielles,
- ✓ Les blocs d'entrée/sortie IOB (*Input/Output Blocks*) qui assurent l'interface avec l'extérieur du circuit,
- ✓ Les matrices d'interconnexions (*Programmable Interconnect*) qui permettent d'interconnecter les blocs.

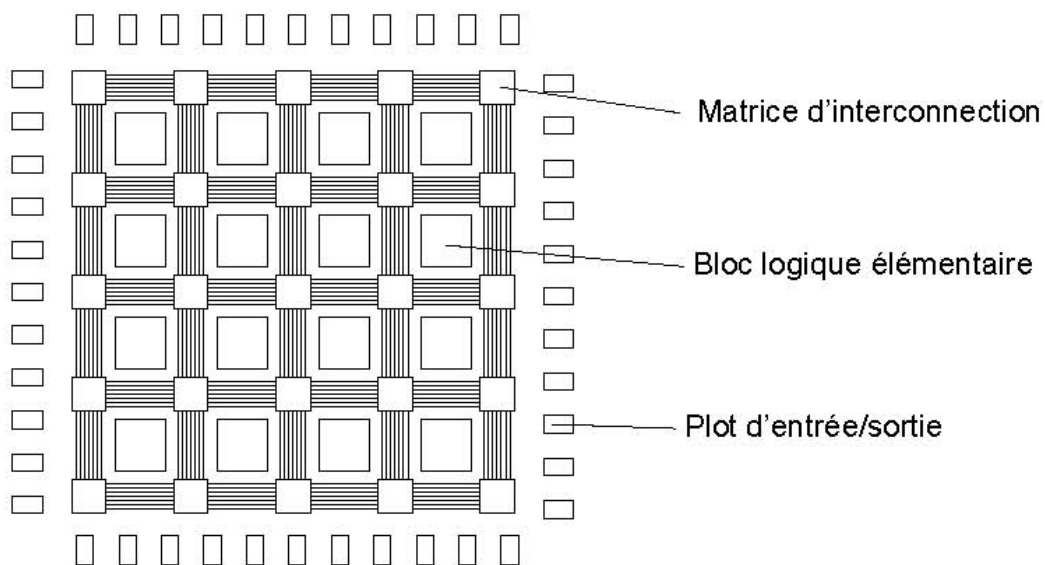


Figure 2.7. Structure interne du circuit configurable [XILINX\_2000].

#### 2.4.1.1. Les blocs logiques CLB (*Configurable Logic Blocs*):

C'est l'unité fondamentale du bloc logique qui fournit des éléments utilitaires pour la logique combinatoire et la logique séquentielle, y compris les éléments de stockage de base. Chaque CLB est composé de quatre slices répartis sur deux colonnes, chaque slice étant composé à son tour de deux cellules logiques (*Logic Cell*) disposées en colonne (Figure 2.8). Cette organisation en colonnes permet d'implémenter efficacement des additionneurs grâce à des chaînes de propagation de retenues rapides.

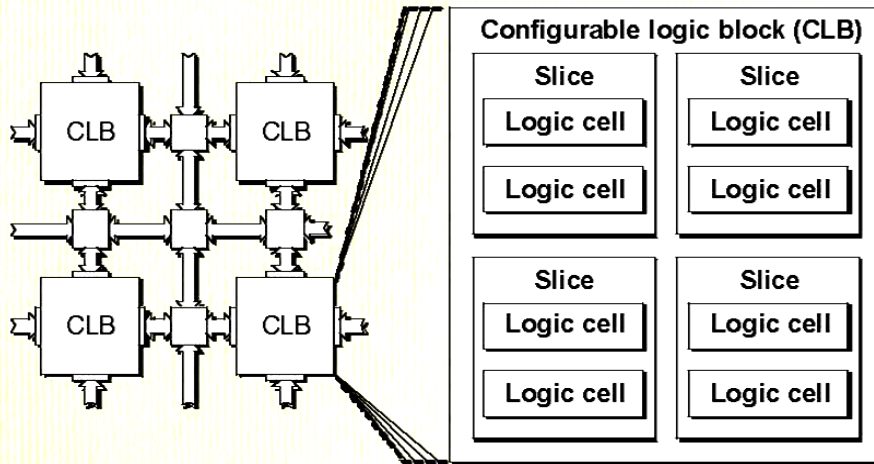


Figure 2.8. Un bloc CLB (*Configurable Logic Blocs*) [XILINX\_2000].

Les cellules logiques d'un FPGA sont généralement constituées d'une partie calculatoire et d'une partie de mémorisation (Figure 2.9). La partie calculatoire est assurée par des générateurs de fonctions, appelés LUT (*Look Up Table*) qui contiennent, après configuration, la table de vérité de la fonction logique qu'elles doivent réaliser, la LUT est donc une petite mémoire de  $2^n$  mots de 1 bit, adressée par n bits provenant de la matrice de routage. Quant à la partie de mémorisation, il s'agit de bascules D (*flip-flop*) contrôlées par un signal horloge, ces cellules disposent par ailleurs de chaînes de retenue (*Carry*) [DETREY\_2007].

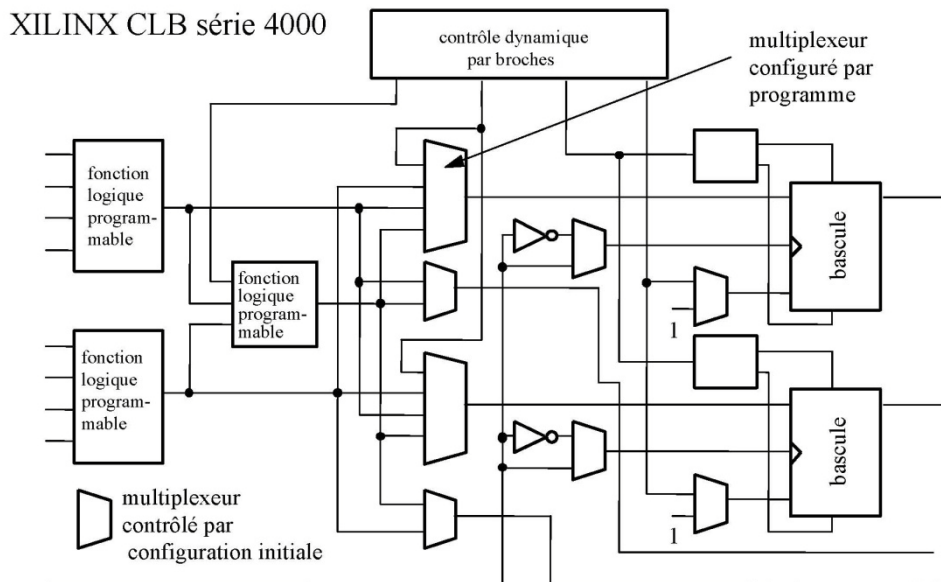


Figure 2.9. Schéma d'une cellule logique (XC4000 de XILINX) [XILINX\_1999].

2.4.1.2. Les blocs d'entrée/sortie IOB (*Input/Output Blocks*) :

Les blocs d'entrée/sortie sont positionnés à la périphérie du circuit FPGA. Ces IOB permettent l'interface entre les broches du composant FPGA et la logique interne développée

à l'intérieur du composant (Figure 2.10) et Chaque IOB peut être configuré en entrée, en sortie, en signal bidirectionnel ou être inutilisé (état haute impédance).

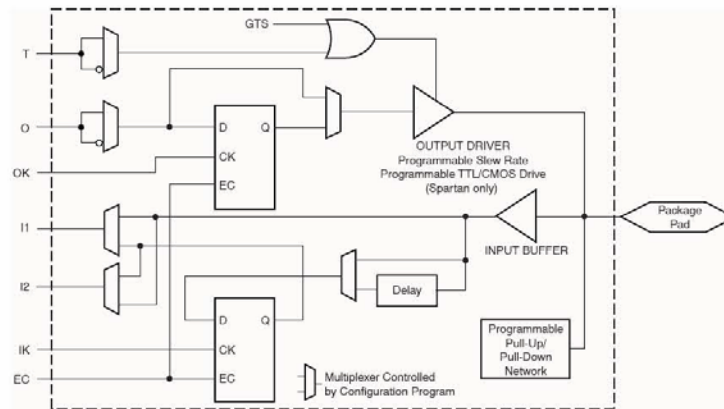


Figure 2.10. Schéma d'un bloc d'entrée/sortie (SPARTAN) [XILINX\_2008].

#### ▪ Configuration en entrée :

Premièrement, le signal d'entrée traverse un buffer qui, selon sa programmation, peut détecter soit des seuils TTL soit des seuils CMOS. Il peut être routé directement sur une entrée directe de la logique du circuit FPGA ou sur une entrée synchronisée. Cette synchronisation est réalisée à l'aide d'une bascule du type D, le changement d'état peut se faire sur un front montant ou descendant. De plus, cette entrée peut être retardée de quelques nanosecondes pour compenser le retard pris par le signal d'horloge lors de son passage par l'amplificateur. Le choix de la configuration de l'entrée s'effectue grâce à un multiplexeur (*Program Controlled Multiplexer*). Un bit positionné dans une case mémoire commande ce dernier.

#### ▪ Configuration en sortie :

Nous distinguons les possibilités suivantes :

- ✓ Inversion ou non du signal avant son application à l'IOB ;
- ✓ Synchronisation du signal sur des fronts montants ou descendants d'horloge ;
- ✓ Mise en place d'un "pull-up" ou "pull-down" (des résistances) dans le but de limiter la consommation des entrées/sorties inutilisées ;
- ✓ Chaque sortie peut délivrer un courant de 12 mA. Ainsi toutes ces possibilités permettent au concepteur de connecter au mieux une architecture avec les périphériques extérieurs.

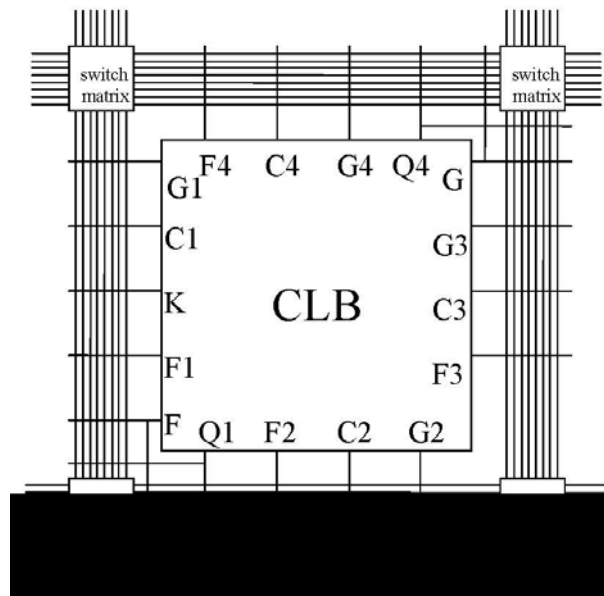
#### 2.4.1.3. Les interconnexions programmables (*Programmable Interconnect*):

Dans les FPGA, on trouve des connexions internes, ils sont composés de segments métallisés et des matrices programmables qui sont réparties sur la totalité du circuit, positionnées horizontalement et verticalement entre les divers CLB. Ces matrices programmables permettent les connexions entre les diverses lignes, qui sont assurées par des transistors MOS dont l'état est contrôlé par des cellules de mémoire vive (RAM).

Le rôle de ces interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible. Pour parvenir à cet objectif, XILINX propose trois sortes d'interconnexions selon la longueur et la destination des liaisons [TARIGHT\_1999]. Nous disposons :

- ✓ d'interconnexions à usage général,
  - ✓ d'interconnexions directes,
  - ✓ de longues lignes.
- **Les interconnexions à usage général :**

Cette configuration fonctionne en une grille, de cinq segments métalliques verticaux et quatre segments horizontaux positionnés entre les rangées et les colonnes de CLB et de l'I/OB (Figure 2.11).



**Figure 2.11. Connexion à usage général.**

À chaque intersection on trouve des matrices de commutation (*Switch Matrix*) qui permettent de raccorder les segments entre eux selon diverses configurations (Figure 2.12); on trouve aussi des buffers implantés en haut et à droite de chaque matrice de commutation dans le but d'éviter que les signaux traversant les grandes lignes ne soient affaiblis. Ces interconnexions sont utilisées pour relier un CLB à n'importe quel autre.

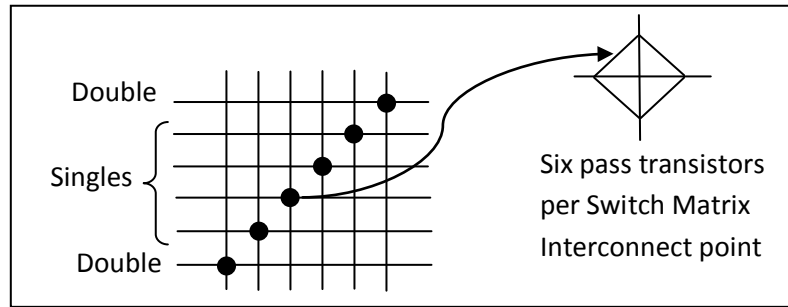


Figure 2.12. Concept d'une matrice de commutation.

▪ Les interconnexions directes :

Ces interconnexions sont des lignes qui permettent des liaisons directes entre les CLB et les IOB, ou entre les différents blocs logiques (Figure 2.13).

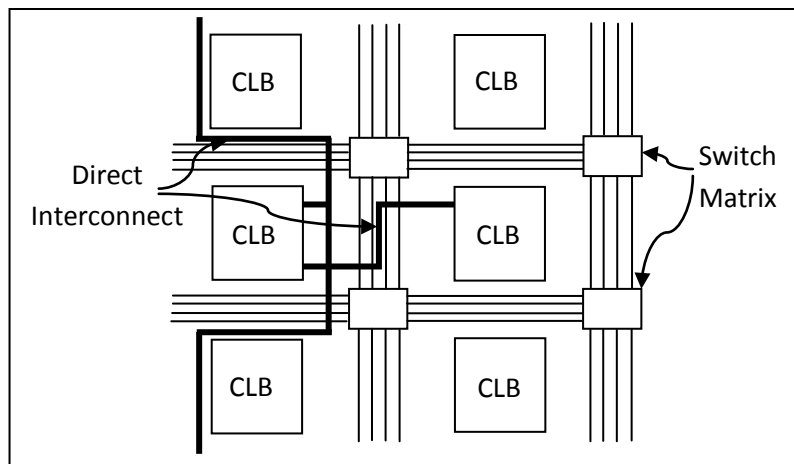


Figure 2.13. Les interconnexions directes.

▪ Les longues lignes :

Ces interconnexions sont de longs segments métallisés qui permettent la transmission des signaux entre les différents éléments avec un minimum de retard dans le but d'assurer un synchronisme aussi parfait que possible (Figure 2.14).

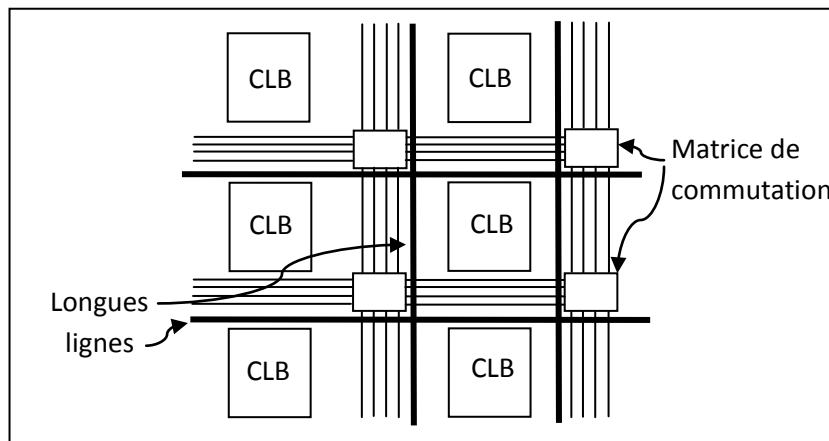


Figure 2.14. Les longues lignes.

L'**horloge** est un élément essentiel pour le bon fonctionnement d'un système électronique. Les circuits FPGA sont prévus pour recevoir une ou plusieurs horloges. Des entrées peuvent être spécialement réservées à ce type de signaux, ainsi que des ressources de routage spécialement adaptées aux transports d'horloges, sur de longues distances.

L'horloge est **un oscillateur à quartz**, placé dans un angle de la puce, il peut être activé lors de la phase de programmation pour réaliser un oscillateur. Il utilise deux IOB voisins, pour réaliser l'oscillateur. Cet oscillateur ne peut être réalisé que dans un angle de la puce où se trouve l'amplificateur prévu à cet effet. Il est évident que si l'oscillateur n'est pas utilisé, les deux IOB sont utilisables au même titre que les autres IOB.

#### **2.4.2. Le réseau mémoire SRAM :**

Les FPGA ont changé et ne sont plus seulement utilisés pour des applications du type "logique de glue" (*Glue Logic*), comme ce fut le cas à leurs débuts, mais aussi pour des applications plus importantes qui demandent souvent des capacités de stockage (comme le traitement d'images). La nécessité d'intégrer des blocs de mémoires directement dans l'architecture des FPGA est vite devenue cruciale. De cette façon les temps d'accès à la mémoire sont diminués puisqu'il n'est plus nécessaire de communiquer avec des éléments extérieurs au circuit.

La programmation du circuit FPGA, appelé aussi LCA (*Logic Cell Array*), consistera en l'application d'un potentiel adéquat sur la grille de certains transistors à effet de champ servant à interconnecter les éléments des CLB et des IOB, afin de réaliser les fonctions souhaitées et d'assurer la propagation des signaux. Ces potentiels sont tout simplement mémorisés dans le réseau mémoire SRAM.

La programmation d'un circuit FPGA est volatile, la configuration du circuit est donc mémorisée sur la couche réseau SRAM et stockée dans une ROM externe. Un dispositif interne permet à chaque mise sous tension de charger la SRAM interne à partir de la ROM.

### **2.5. Critères de choix pour la réalisation d'un circuit logique:**

Le concepteur d'un système numérique doit faire un ou plusieurs choix parmi les circuits qui ont été présentés pour la réalisation de circuit logique; les critères le plus souvent utilisés sont présentés ci-après :

#### **2.5.1. Coût de développement et fabrication :**

C'est le coût des dépenses engagées pour concevoir le système et réaliser les outils nécessaires à sa fabrication et son test. Ce critère intervient dans le choix du circuit, que le nombre de pièces à réaliser est faible, ce qui favorise l'utilisation des processeurs, les différents circuits programmables (SPLD, CPLD, FPGA) que les ASIC (surtout les *Full Custom*).

L'avantage principal des FPGA est un faible coût pour une production à faible unité, ou pour le développement de prototypes, car ce dernier est reprogrammable indéfiniment. Par

contre, il devient d'un coût plus important qu'un circuit ASIC pour une production de plus grandes séries (Figure 2.15).

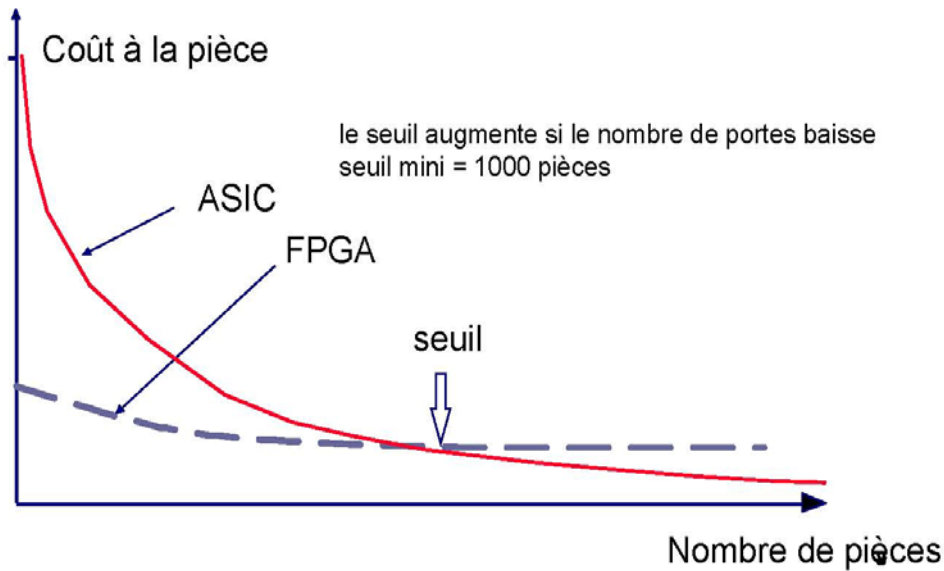


Figure 2.15. Coût des FPGA par rapport aux ASIC [KARABERNOU\_2009].

2.5.2. Temps de développement :

Si les circuits ASIC présentent des avantages évidents en termes de capacité d'intégration et de vitesse de calcul, ils présentent par contre des inconvénients d'une part en termes de temps de développement, car la réalisation d'un circuit ASIC nécessite une phase de fabrication, qui peut être de plusieurs semaines (Figure 2.16) et d'autre part en termes de coût de développement qui sont relativement importants.

Le temps de fabrication à l'aide d'un circuit FPGA se résume à sa programmation ce qui est négligeable.

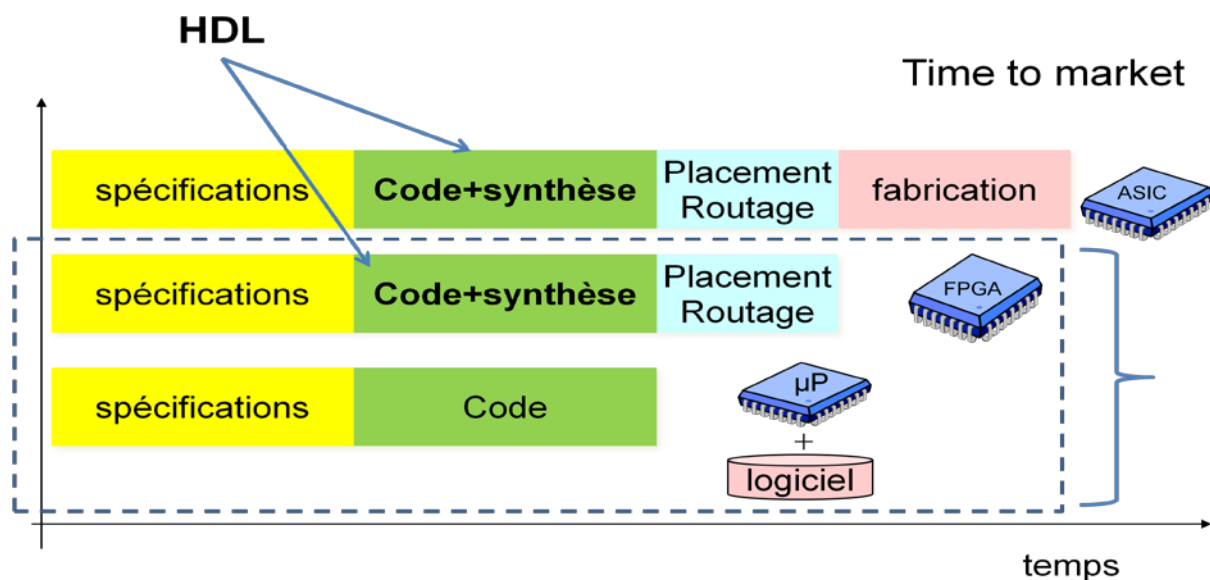


Figure 2.16. Temps de conception [KARABERNOU\_2016].

### 2.5.3. Souplesse d'utilisation :

Favorise les circuits programmables (SPLD, CPLD, FPGA) dont on peut modifier plus facilement des fichiers que des circuits.

### 2.5.4. Taille :

Il y a une forte dépendance entre la taille du système et la densité d'intégration. L'augmentation de la densité d'intégration produit des systèmes de taille réduite. La figure suivante représente la densité d'intégration (Figure 2.17) dans les différentes familles des circuits logiques.

Les ASIC sont les mieux adaptés puisqu'ils sont totalement adaptés à l'application (plus particulièrement les *Full custom* et les *Standard Cell*).

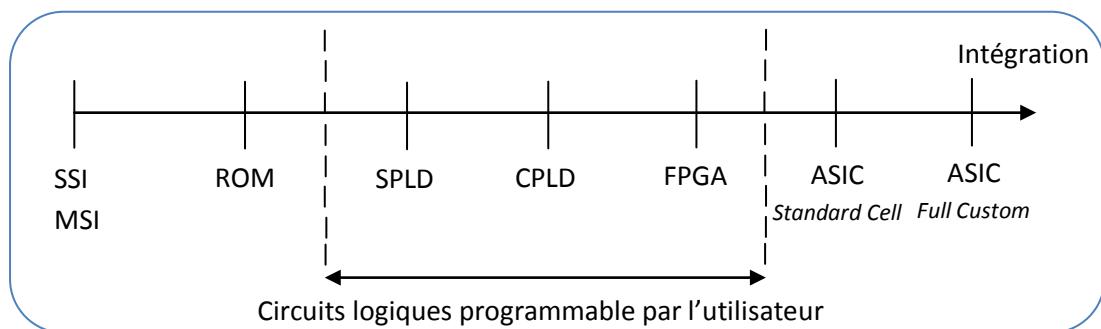


Figure 2.17. L'intégration dans les circuits logiques.

### 2.5.5. Consommation :

Critère particulièrement sensible dans les applications possédant une alimentation autonome. Il conduit à favoriser des solutions ASIC pour les mêmes raisons que le critère précédent [DOUILLARD\_2005].

### 2.5.6. Vitesse de fonctionnement :

Les CPLD sont des composants pour la plupart reprogrammables électriquement ou à fusibles, peu chers et très rapides (fréquence de fonctionnement élevée) mais avec une capacité fonctionnelle moindre que les FPGA.

### 2.5.7. Capacité mémoire :

Les FPGA à SRAM contiennent des mémoires pour stocker leur configuration. La plupart des familles récentes offrent à l'utilisateur, la possibilité d'utiliser certaines de ces mémoires en tant que telles [WEBER\_2000].

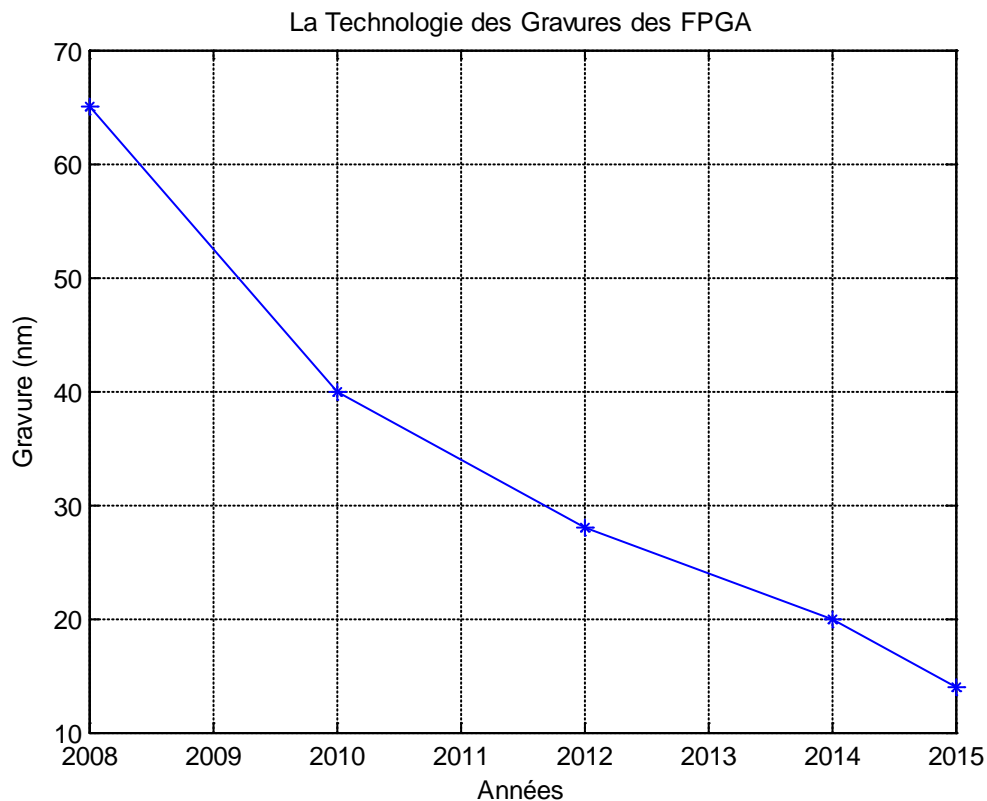
## 2.6. Notions sur les FPGA actuelles :

La part de marché des FPGA dans le marché global des circuits matériels pour l'électronique numérique ne cesse d'augmenter : les évolutions technologiques, les domaines d'application



### 2.6.1. Les technologies avancées des FPGA :

Les FPGA utilisent les technologies les plus avancées disponibles à tout moment donné (Figure 2.18).



**Figure 2.18. Évolution des technologies d'intégrations des FPGA.**

Concevoir un ASIC dans ces technologies permettrait d'obtenir un système extrêmement performant, mais au prix d'un investissement colossal. Il faut, en effet, une équipe de conception importante, des outils coûteux et complexes, plusieurs mois de conception et enfin un budget important pour la fabrication. De plus, les délais de fabrication chez le fondeur peuvent être importants (quelques mois). Le temps de conception et vérification d'une solution FPGA est beaucoup plus court qu'en ASIC et nécessite moins d'efforts et de personnel. Bon nombre de détails techniques sont réglés par le fabricant du FPGA.

### 2.6.2. Les fabricants et parts du marché des FPGA :

Parmi les principaux fabricants de FPGA dans le monde (Figure 2.19): Xilinx (n°1 du marché des FPGA, 53% de parts de marché en 2016), Altera (n°2 du marché des FPGA, 36% de parts de marché en 2016), Actel, Lattice, QuickLogic [DILLIEN\_2017].

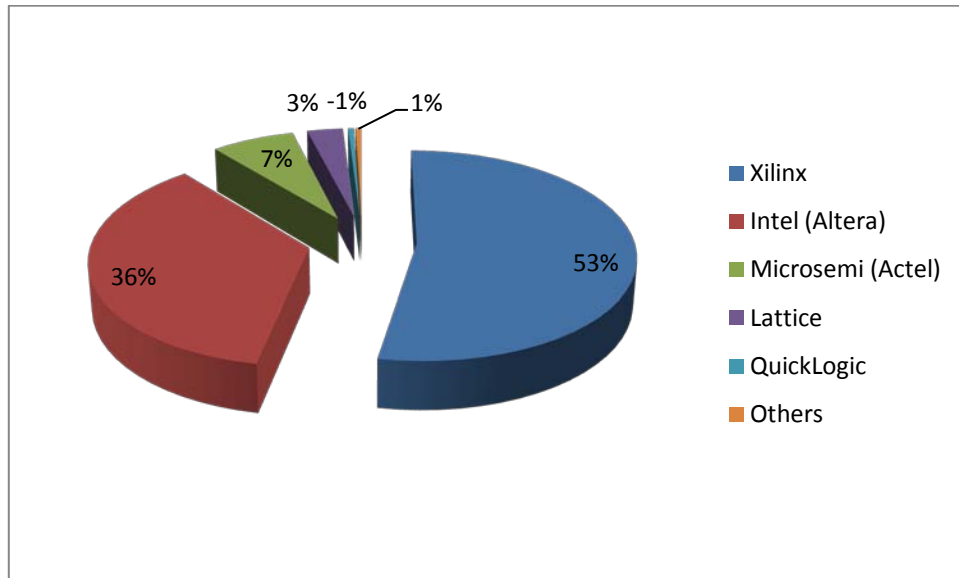


Figure 2.19. Les principaux fabricants des circuits FPGA.

La bonne santé du marché des composants logiques programmables, les FPGA, très populaires dans de nombreux domaines d'application dont l'embarqué, ne se dément pas. Selon la société d'analystes **Transparency Market Research** (le marché mondial des FPGA) [GAUTHIER\_2013], ce marché estimé en 2012 à 5.08 milliards de dollars, pourraient atteindre 8.95 milliards de dollars en 2019, soit une croissance annuelle de 8.5% (Figure 2.20).

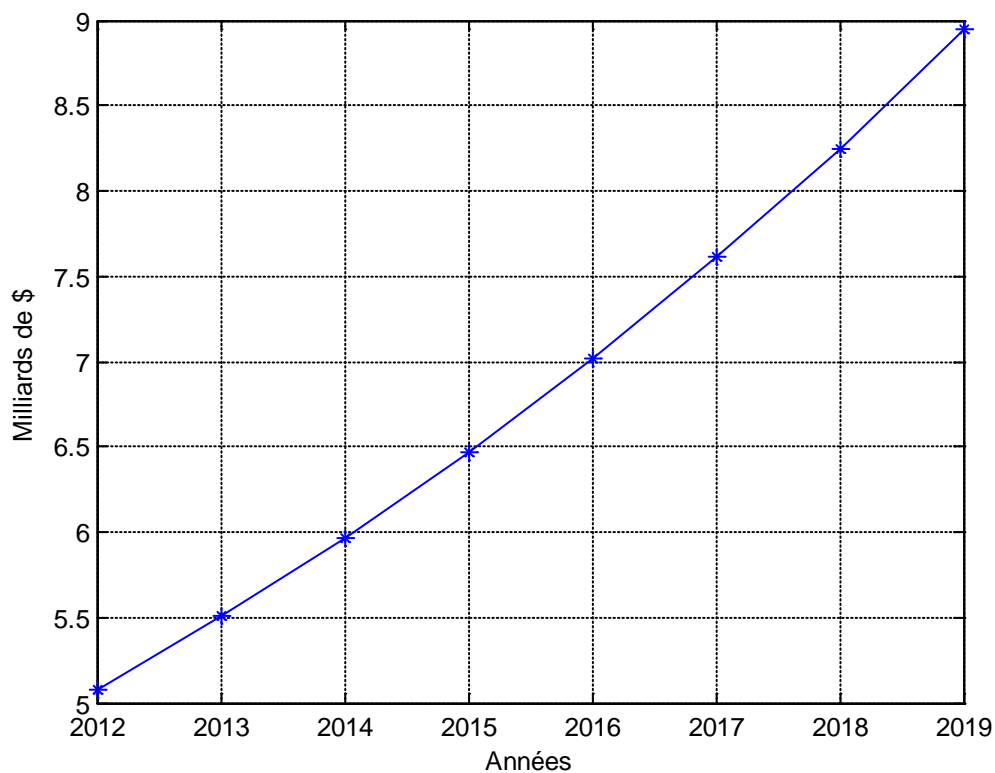


Figure 2.20. Le marché des FPGA.

### 2.6.3. Les domaines d'application des FPGA:

Les FPGA fournissent les très hautes performances du matériel mais avec une utilisation simple et rapide par configuration logicielle. Pour ce fait, nombreux domaines favorisent l'utilisation de ces circuits comme les systèmes embarqués, télécommunication, le traitement du signal et de l'image, l'imagerie médicale, les capteurs biomédicaux, les réseaux sans fil, le cryptage, dispositifs de sécurité et le prototypage des circuits etc... Nous allons définir quelques domaines d'application.

#### 2.6.3.1. Domaine des systèmes embarqués :

Nous trouvons les SoC (*System on Chip*) qui sont des circuits ASIC intègrent des processeurs, mémoires, interfaces et parfois des FPGA [POUSSIER\_2002].

Intel (le premier fabricant mondial de semi-conducteur) avait dévoilé être devenu le partenaire de fonderie d'Altera pour ses circuits les plus avancés en février 2013. Un an plus tard en mars 2014, Altera et Intel annonçaient leur collaboration au développement de composants multi-puces, associant la technologie de logique programmable d'Altera à l'expertise d'Intel dans l'assemblage et le packaging avancé. Les deux compères souhaitaient développer des **composants multi-puces** intégrant, dans un même boîtier, des FPGA 14 nm Stratix10 et des systèmes sur une puce (SoC) avec d'autres types de composants (Dram, Sram, Asic, processeurs et composants analogiques). Ces composants visent les applications hautes performance en termes de taille mémoire et dissipation thermique pour les marchés des communications, de l'informatique haute performance et de la défense [ETATS\_2015].

L'approche SoC répond aux besoins de performances et d'intégration mais :

- Elle s'adapte difficilement aux besoins des systèmes.
- Elle reste réservée aux grands volumes de production.
- La fabrication et le test sont des étapes longues et coûteuses.

Afin de répondre à ces problèmes, certains modèles de FPGA intègrent directement un ou plusieurs cœurs complets de processeurs en faisant naissance aux systèmes programmables sur puce SoPC (*System on a Programmable Chip*). Cette intégration peut se faire soit d'une façon logicielle (*SoftCore*) dont le processeur est décrit en langage de description matérielle comme le MicroBlaze de Xilinx et le NiosII d'Altera, ou d'une manière matérielle (*HardCore*) où le processeur est gravé sur le silicium au sein du FPGA comme ARM Cortex-A9 d'Altera [ARLOT\_12/2014].

Altera affirme être le premier spécialiste de la logique programmable à câbler dans le silicium de certains de ses FPGA des opérateurs de calcul en virgule flottante compatibles IEEE 754. Ces blocs durcis de DSP à virgule flottante sont intégrés dans les SoCs et FPGA Arria10 d'Altera gravés en technologie 20 nm. Cette dernière permet notamment aux Arria10 d'atteindre une performance supérieure à ceux offerts par les premiers DSP, les processeurs CPU et les processeurs graphiques GPU [ARLOT\_04/2014].

### 2.6.3.2. Domaine de télécommunication :

Selon Transparency Market Research (le marché mondial des FPGA), le déploiement des technologies 3G et LTE (*Long Term Evolution* : est une évolution des normes de téléphonie mobile) dans le monde, avec des demandes en bande passante élevée, va tirer ce marché, avec des taux de croissance moyens annuels supérieurs à 9% entre 2013 à 2019, rien que pour ce domaine particulier. Le secteur des écrans pour les SmartPhone, Tablettes et Phablettes, terminaux dotés de fonctions tactiles de plus en plus évoluées, sera également très favorable à la progression du marché des FPGA [GAUTHIER\_2013].

### 2.6.3.3. Domaine médical :

La recherche médicale ne peut avancer aussi rapidement que la technologie qui la soutient. La société Actel annonce plusieurs succès dans les applications médicales, allant de l'appareil de surveillance et de tests à domicile en passant par l'équipement clinique jusqu'aux applications d'imagerie [PISTES\_2009].

Dans les appareils de tests et de surveillance à domicile portables et alimentés par batterie comme les pompes à perfusion ou à insulines, les tensiomètres et les glucomètres, la société Actel intègre ses FPGA IGLOO de 5 $\mu$ W monopuce Fusion à signal mixte (FPGA à base de mémoire Flash), qui peut assurer la gestion thermique, la consommation et l'enregistrement des données.

Beaucoup d'équipements cliniques, tels que l'équipement pour le diagnostic de laboratoire, les défibrillateurs externes automatiques (AED) et les machines d'hémodialyse, peuvent profiter de l'intégration et des capacités que peuvent apporter les FPGA à signaux mixtes. Mindray Medical (le plus grand fabricant chinois dans les équipements médicaux) a choisi les FPGA ProASIC3 à base de Flash d'Actel pour ses systèmes portables de surveillance de patient car ses FPGA étaient d'ultra basse consommation, performants, fiables et avaient une sécurité renforcée.

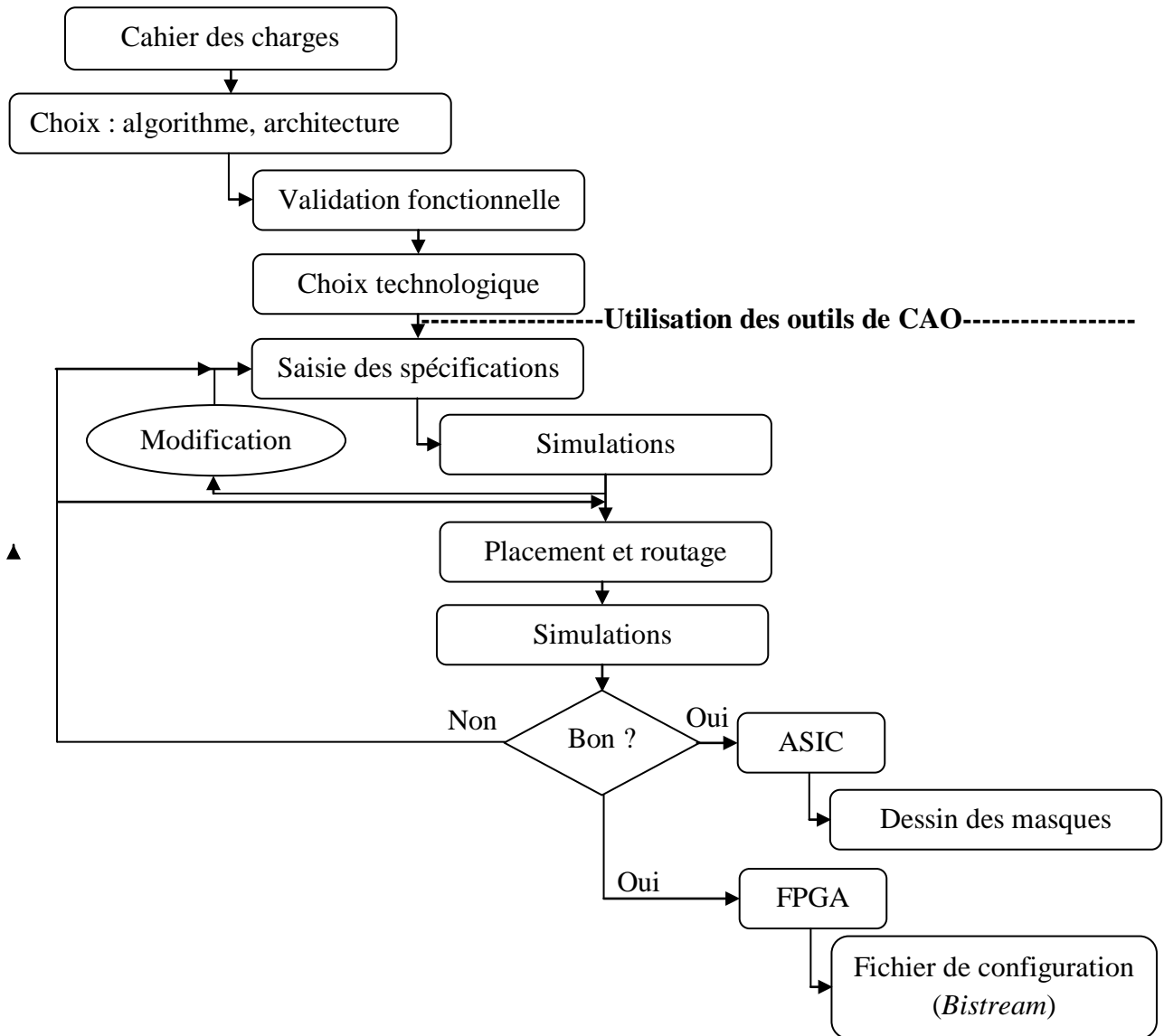
Les FPGA ont rapidement gagné en popularité pour les applications médicales, notamment en imagerie médicale. Les FPGA sont particulièrement utilisés dans la détection et la construction de l'image. Nous trouvons comme exemple le système d'imagerie médicale par OCT 3D (la tomographie par cohérence optique), c'est un système qui permet de visualiser des images 3Ds en temps réel grâce à une instrumentation modulaire PXI intégrant des circuits FPGA pour traiter et transmettre les signaux collectés à des vitesses très élevées [DEVICE\_2012].

## 2.7. Méthodologie de conception :

Avec l'évolution des circuits numériques dans la densité d'intégration, nous sommes obligés à suivre un flot de conception en utilisant les différents outils de développement.

**2.7.1. Méthode de conception :**

Le flot de conception classique des circuits de hautes densités (CPLD, FPGA, ASIC) est présenté par l'organigramme suivant :



**Organigramme 2.1. La conception des circuits à hautes densités.**

**2.7.1.1. Le cahier des charges :**

Le point de départ est le cahier des charges, qui définit la fonctionnalité du circuit ainsi que les conditions de fonctionnement (telles que la température de fonctionnement, les vibrations et le rayonnement électromagnétique subis...), il doit spécifier aussi les contraintes d'encombrements, de tension d'alimentation, de vitesse de fonctionnement...

**2.7.1.2. La validation fonctionnelle :**

Le concepteur commence par écrire un modèle comportemental (ou fonctionnel) du circuit en choisissant un algorithme et une architecture convenable. Cette étape consiste à identifier

les principales fonctions et faire un découpage en blocs du circuit; cette découpe est généralement réalisée en plusieurs étapes amenant à des descriptions de plus en plus détaillées. La découpe est terminée lorsque chaque fonction élémentaire peut être réalisée à l'aide d'éléments disponibles dans la bibliothèque utilisée pour la synthèse logique (fonctions combinatoires élémentaires, opérateurs arithmétiques, compteurs, registres, automates, ...). Cette bibliothèque est fournie par le fondeur choisi pour la fabrication du circuit. Le but de cette étape est de valider la partie fonctionnelle du cahier des charges en respectant toujours les contraintes du temps et de surface.

### **2.7.1.3. Le choix technologique :**

Ensuite, on va établir le choix technologique c'est-à-dire le choix de la manière de décrire un circuit. Soit on veut faire fabriquer ce circuit avec un ASIC (*Application Specific integrated circuit*), ou bien l'implémenter sur un FPGA (*Field Programmable Gate Arrays*).

### **2.7.1.4. La synthèse :**

L'étape suivante est la synthèse, indépendante de la technologie du circuit cible, contient la saisie de l'application dans un outil de CAO (Conception Assistée par Ordinateur) (cf. § 7.2.1), en tenant compte de la nature de la fonction à réaliser et du composant qui va accueillir cette fonction, car c'est des critères importants pour faire un choix du mode de description qui facilite la conception de ce circuit. On peut définir trois descriptions :

Une description schématique utilise des composants de base (hard macro, composants optimisés) fournie pour un circuit donné ou du moins une famille de ce dernier. On utilisera ce type de description lorsque la nature de la fonction à implanter est très structurée. On peut dire que ce fichier source (schéma) est difficilement portable même si l'ensemble des composants de base utilisés dans les conceptions numériques se limitent à des fonctions restreintes et génériques (bascules, compteurs...), car chaque constructeur propose sa propre librairie de composants et les noms utilisés pour ces composants génériques ne sont en général acceptés que par le compilateur du circuit cible.

Une description HDL utilise un langage du type comportemental permet de rendre le code source indépendant de la cible. Un code HDL normé (VHDL, VERILOG, ...) sera accepté par la plupart des outils (simulateurs, synthétiseurs...).

Une description mixte consiste à définir certains éléments hiérarchiques en schématique et d'autres de manière syntaxique. Il pourra par exemple s'avérer judicieux de définir schématiquement la structure globale d'une conception et de définir de manière comportementale chaque bloc fonctionnel [DUTRIEUX\_1997].

La synthèse aboutit à une description du circuit au niveau logique, couramment appelée : "netlist". Cette étape prend en compte les contraintes imposées par le cahier des charges (surface, temps). La synthèse est validée toujours par une simulation logique qui utilise les caractéristiques temporelles fournies par la bibliothèque.

### 2.7.1.5. La simulation fonctionnelle:

Cette simulation peut être directe ou avec un banc d'essai (*test bench*). La simulation directe permet de charger le composant à simuler dans un simulateur (par exemple : MODELSIM) directement et faire varier entrée par entrée à chaque pas de simulation en définissant le temps qui dépendra de la fréquence du signal d'horloge.

La simulation par un fichier de test (*test bench*) consiste à créer un composant B sans entrées –sorties (Figure 2.21) en déclarant le composant A à simuler et les signaux internes (autant de signaux que les entrées-sorties du composant A). Ce banc d'essai est un module écrit en VHDL qui permet de faire varier les signaux internes connectés aux entrées afin de visualiser les résultats sur une fenêtre des courbes [KARABERNOU\_2009].

Cette dernière méthode de simulation est recommandée car elle est méthodique et elle nous permet de vérifier le fonctionnement de la description à chaque étape de la procédure de développement.

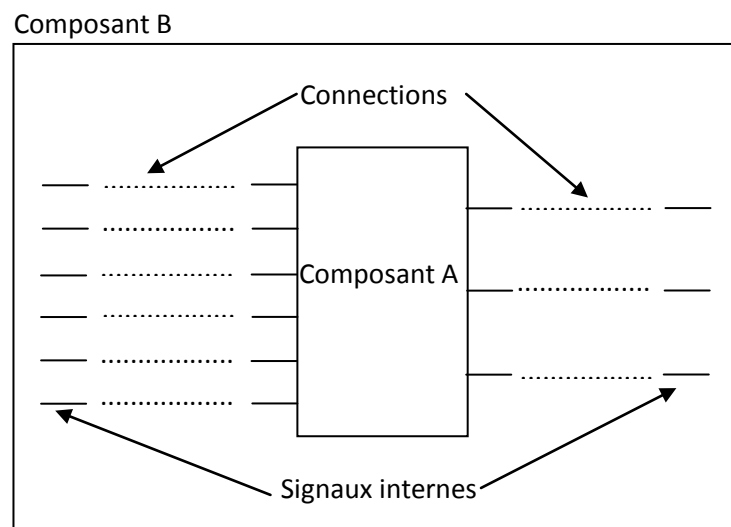


Figure 2.21. Simulation par un banc de test.

### 2.7.1.6. Le placement et le routage :

Reste ensuite à déterminer l'emplacement physique des blocs logiques sur le silicium (placement) et à déterminer les chemins suivis par les interconnexions (routage). Les problèmes du placement et du routage ne sont pas indépendants : la qualité du placement pouvant fortement influencer le routage dans le circuit.

Le placement consiste aussi à affecter les signaux utilisés dans le circuit à ses plots d'entrées/sorties.

### 2.7.1.7. La simulation temporelle :

À l'issue de ces opérations (après simulation), il est possible d'extraire les retards apportés par les connexions, et de modifier la netlist en conséquence. Cette opération s'appelle la rétro-annotation. En effet, à la synthèse, ce sont des valeurs par défaut correspondant à des capacités estimées moyennes qui sont utilisées. Une simulation post-routage utilisant les

retards réels permet de vérifier si le circuit définitif répond bien aux contraintes du cahier des charges. Sinon, il faut refaire la synthèse en tenant compte des résultats de la rétro-annotation. Si les résultats de la rétro-annotation sont parfaits, on va établir l'étape suivante!

### 2.7.1.8. La fabrication ou la programmation du circuit :

Si c'est un ASIC, on réalise le dessin des masques en respectant les règles du dessin afin d'établir la production du circuit. Et si c'est un FPGA un fichier de configuration, contenant l'ensemble des informations relatives à l'implémentation, est produit. Celui-ci peut alors être utilisé pour configurer le circuit programmable, soit à l'aide d'une mémoire non volatile associée au circuit, soit directement à partir d'une interface externe (processeur, bus, liaison parallèle, etc...).

### 2.7.2. Les outils de développement :

Avec l'apparition des circuits logiques programmables du type SPLD, CPLD ou FPGA, l'utilisateur peut créer dans ces derniers toutes les fonctions logiques en utilisant des outils de développement mis à sa disposition par leurs fabricants ; ces outils doivent permettre de passer de la description du comportement d'une fonction logique à son câblage dans le circuit et cela de la manière la plus simple possible (Figure 2.22).

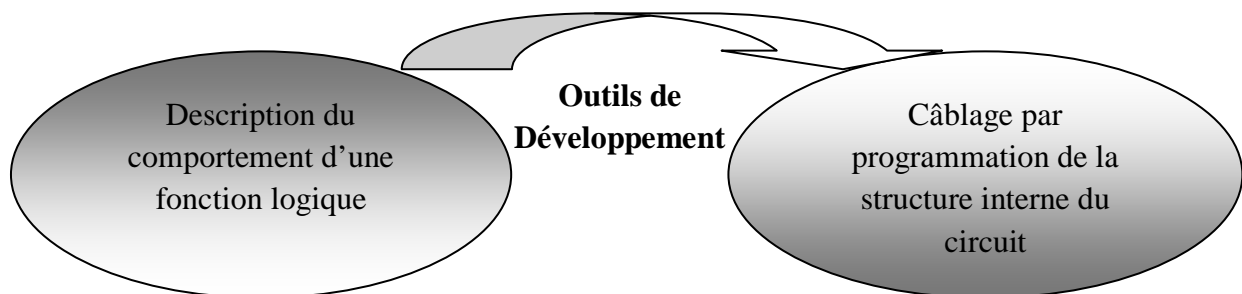


Figure 2.22. L'utilité d'un outil de développement.

#### 2.7.2.1. Les outils de CAO (Conception Assistée par Ordinateur) :

La conception assistée par ordinateur CAO (*Electronic Design Automation* : EDA) est la catégorie des outils servant à la production des systèmes électroniques allant des circuits imprimés jusqu'aux circuits intégrés.

Avec la miniaturisation continue de la technologie des semi-conducteurs, les circuits électroniques sont devenus trop complexes pour être conçus à la main ce qui a résulté la nécessité des outils informatiques adéquats et c'était la naissance des outils de CAO.

Les concepteurs réalisent et testent les circuits sur ordinateur avant de lancer la fabrication. On peut distinguer trois types d'outils de CAO selon les modes de descriptions trouvés dans ces derniers :

- Un outil de CAO utilisé que pour la synthèse et la simulation (comme MODELSIM).
- Un outil de CAO qui utilise le mode schématique et textuel (comme ISE de XILINX).



- Un outil de CAO qui peut utiliser le mode schématique, textuel et permet même le dessin du masque (*layout*) (comme CADENCE).

### 2.7.2.2. Les principaux langages de description matérielle :

#### Le langage VHDL :

Développé dans les années 80 aux États-Unis, le langage VHDL (*Very high speed integrated circuits Hardware Description Language*) est ensuite devenu une norme IEEE numéro 1076 en 1987. Révisée en 1993 pour supprimer quelques ambiguïtés et améliorer la portabilité du langage, cette norme est vite devenue un standard en matière d'outils de description de fonctions logiques [BLOTIN\_2005]. À ce jour, on utilise le langage VHDL pour :

- concevoir des ASIC,
- programmer des composants programmables du type PLD, CPLD et FPGA,
- concevoir des modèles de simulations numériques ou des bancs de tests.

Le langage VHDL assure la portabilité des conceptions, permet de synthétiser des fonctions logiques complexes à partir d'une description concise. Il est écrit indépendamment d'une technologie particulière ou d'une chaîne de conception, facilite le passage entre les différentes technologies utilisables qui évoluent sans cesse.

Le VHDL a pour autre avantage la possibilité de décrire des systèmes très complexes en quelques lignes de code. Cependant, la synthèse de ces fichiers devient de plus en plus difficile et un code VHDL doit être optimisé pour un circuit cible. Dans la pratique, une fois ces modules développés, on les intègre dans une bibliothèque et les réutilise en cas de besoin ultérieur. Pour des circuits plus importants, cela peut devenir très compliqué à développer, surtout pour des entreprises de petite à moyenne importance. Certaines entreprises de circuits ont alors proposé de vendre leur propriété intellectuelle sous forme de code VHDL et de *tests-benchs* prêts à l'emploi. L'utilisateur n'a alors qu'à assembler des modules sans avoir à écrire le code interne [GUEX\_1998].

#### VERILOG :

VERILOG est un langage de description de matériel, c'est-à-dire un langage utilisé pour décrire un système numérique matériel. Il a été inventé par *Gateway Design Automation Inc* en 1985, et en 1990, la société de CAO Cadence rachète Gateway. Elle décide de rendre les spécifications de VERILOG publiques, qui deviennent un standard IEEE en 1995 (IEEE Std. 1364-1995).

L'IEEE a créé une nouvelle version du langage, VERILOG 2001 puis VERILOG 2005 à cause des lacunes de VERILOG 1995. Ces versions du langage sont maintenant acceptées par la quasi-totalité des outils de l'industrie.

VERILOG peut modéliser un système par n'importe quelle vue, structurelle ou comportementale, à tous les niveaux de description, de plus il peut servir non seulement à

simuler un système mais aussi à le synthétiser, c'est-à-dire être transformé par des logiciels adaptés (synthétiseurs) en une série de portes logiques prêtes à être gravées sur du silicium.

VERILOG est l'un des trois grands langages de description de matériel, utilisés dans l'industrie, avec VHDL et SystemC. Il combine deux aspects :

- la simulation : il permet de décrire l'enchaînement d'événements ;
- description par combinaison d'éléments (modules, expressions, portes logiques...), ce qui permet de synthétiser des circuits.

La structure du langage VERILOG permet de décrire les entrées et les sorties de modules électroniques, pour définir des portes logiques virtuelles. La combinaison de modules permet de réaliser des schémas électroniques virtuels complexes qu'il est alors possible de tester dans un programme de simulation [WIKIPEDIA\_2014].

#### Le systemC :

SystemC est, comme VERILOG et VHDL, un langage de description de matériel. C'est le fruit de contributions de plusieurs sociétés. En 1989, *Synopsys* met son outil commercial SCENIC dans le domaine libre, et crée la version 0.9 de SystemC. Une première contribution de *Frontier Design* donne lieu à la version 1.0, et une autre de *CoWare* aboutit en 2000 à la version 1.1, première version officielle de SystemC. L'OSCI (*Open SystemC Initiative*) est alors créé, rassemblant une multitude de sociétés et laboratoires de recherche. Cette organisation est en charge de diffuser, promouvoir et rédiger les spécifications de SystemC.

Les spécifications de SystemC ont été étendues en 2001 à la modélisation de systèmes abstraits (de très haut niveau, avant partitionnement matériel / logiciel), aboutissant à la version 2.0. En 2005, l'IEEE a approuvé une version de référence de SystemC, appelée IEEE1666-2005 et correspondant à la version 2.2.0 de SystemC, qui est l'actuelle version stable.

SystemC a pour objectif de modéliser des systèmes numériques matériels et logiciels à l'aide de C++. Il permet donc de modéliser non seulement des systèmes matériels, mais aussi des systèmes logiciels, mixtes ou non partitionnés (où on ne sait pas encore ce qui sera fait en logiciel, et ce qui sera fait en matériel) [WIKIPEDIA\_2015].

## 2.8. Conclusion:

Les circuits FPGA permettent aujourd'hui de concevoir sur un même circuit intégré des applications en Co Design dans laquelle le matériel et le logiciel se côtoient. Ce Co Design permet de fusionner la flexibilité de la logique programmée sur microprocesseur et les performances de la logique câblée sur circuits programmables.

Avec ces évolutions c'est tout un nouveau domaine de l'électronique numérique qui s'est ouvert. Aujourd'hui les FPGA sont utilisés dans tous les domaines, des systèmes embarqués aux systèmes de communication, ils sont au cœur d'un important champ de recherches académique et industrielle.

Donc nous allons exhiber dans le prochain chapitre l'utilité des composants FPGA dans le domaine médical. Pour cela, nous avons choisi de réaliser et d'implémenter un classifieur neuronal dans le circuit reconfigurable afin de détecter une pathologie dans le cœur en temps réel.



*Chapitre 3*

*La Conception Matérielle du  
Classifieur Neuronal.*

### 3.1. Introduction :

Dans le cadre des recherches effectuées sur la classification neuronale des données médicales et l'importance d'une détection d'une pathologie en temps réel, nous avons choisi la conception matérielle d'un classifieur neuronal pour la reconnaissance des arythmies cardiaques afin d'être implémenté sur le circuit reconfigurable FPGA.

Le but principal de cette thèse est de concevoir puis d'implémenter une architecture matérielle sur un système numérique à base de FPGA, cette dernière peut être utilisée dans un appareil médical permettant la détection des arythmies cardiaques en temps réel, afin de réaliser un système d'aide au diagnostic, qui consiste à l'extraction de connaissances dans de grands volumes de données pour en extraire des informations pertinentes. Ce système est destiné à soutenir l'expert dans sa prise de décision. L'architecture étudiée s'appuie sur des contraintes d'encombrement minimales et de souplesse maximale.

Dans ce chapitre, nous proposons dans un premier temps un classifieur neuronal sous l'environnement Matlab en montrant ses différentes parties, puis présentons la conception matérielle globale du classifieur. Ensuite, nous détaillerons les différents blocs de notre architecture et nous mettrons leurs rôles en évidence.

### 3.2. Modèle du classifieur neuronal proposé :

Les neurones artificiels sont utilisés dans plusieurs applications, sous forme d'architectures variées afin de doter un système quelconque d'une certaine intelligence. Cette dernière est acquise après un processus d'apprentissage artificiel [BENEDIC\_2007].

Les réseaux de neurones sont une formulation mathématique simplifiée des neurones biologiques. Ils ont la capacité de mémorisation, de généralisation et surtout d'apprentissage qui est le phénomène le plus important. Dans ce type de réseau de neurones, chaque neurone a d'abord le rôle de sommer les différentes entrées pondérées par leurs poids respectifs selon l'expression suivante :

$$a(i) = \sum_{j=1}^n W(i,j) * p(j) \quad (3.1)$$

Puis faire passer la somme obtenue dans une fonction  $F$  linéaire ou non linéaire et finalement, de transmettre le résultat aux neurones de la couche suivante. Le modèle mathématique d'un neurone  $i$  est donné par l'équation suivante :

$$T(i) = F(b + \sum_{j=1}^n W(i,j) * p(j)) \quad (3.2)$$

Les variables  $n$ ,  $T(i)$ ,  $F$ ,  $p(j)$ ,  $b$  et  $W$  désignent respectivement le nombre d'entrées du réseau de neurones, le vecteur de sortie du réseau, la fonction d'activation, le vecteur des entrées du réseau de neurones, la valeur du biais et la matrice des poids. Nous présentons dans la (Figure 3.1) la structure d'un neurone simple.

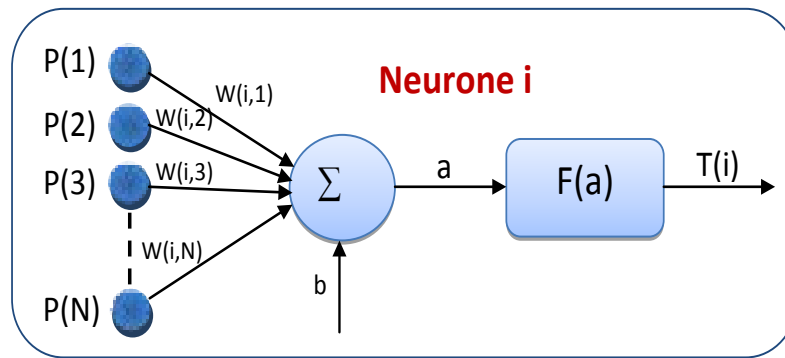


Figure 3.1. Structure d'un neurone formel.

### 3.2.1. Le perceptron multicouche :

D'une façon générale, un perceptron multicouche (*Multi Layer Perceptron*) contient une couche d'entrée, une ou plusieurs couches cachées, chargées des calculs possédant une fonction d'activation et une couche de sortie. La couche d'entrée représente toujours une couche virtuelle associée aux entrées du système, elle ne contient aucun neurone tandis que les couches suivantes sont des couches de neurones. La couche de sortie correspond aux sorties du système.

Dans notre travail, nous utilisons une structure très classique de réseau neuronal [ANAND\_1992] le perceptron multicouche décrit en (Figure 3.2). Nous avons utilisé un réseau de neurones pour la reconnaissance des arythmies cardiaques CLN, qui contient 7 neurones dans la couche d'entrée (correspond aux 7 descripteurs qui caractérisent un battement cardiaque), 3 neurones dans la couche cachée et un neurone dans la couche de sortie (voir section 3.2.4). Ce modèle CLN (7-3-1) était obtenu après plusieurs tests [CHIKH\_2004].

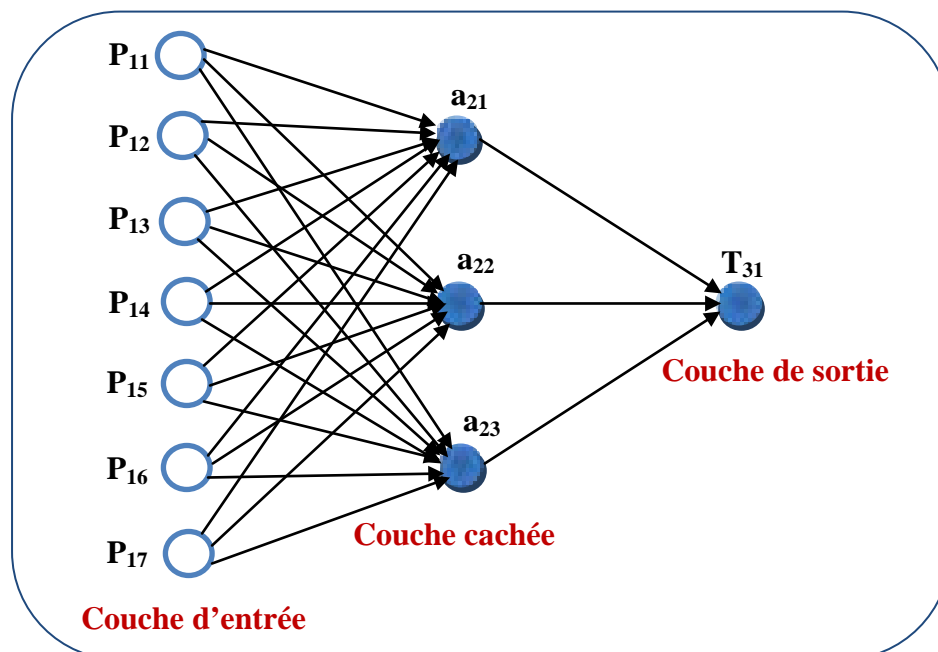


Figure 3.2. Structure d'un perceptron à deux couches.

Dans le perceptron multicouche, chaque neurone à l'intérieur d'une couche reçoit des signaux provenant de la couche précédente, effectue un calcul et transmet le résultat à la couche suivante, si elle est présente. Il n'existe pas d'interconnexions entre les neurones situés à l'intérieur d'une même couche : les activations des différents neurones sont seulement propagées de la couche d'entrée vers la couche de sortie à travers tous les neurones constitutifs du réseau.

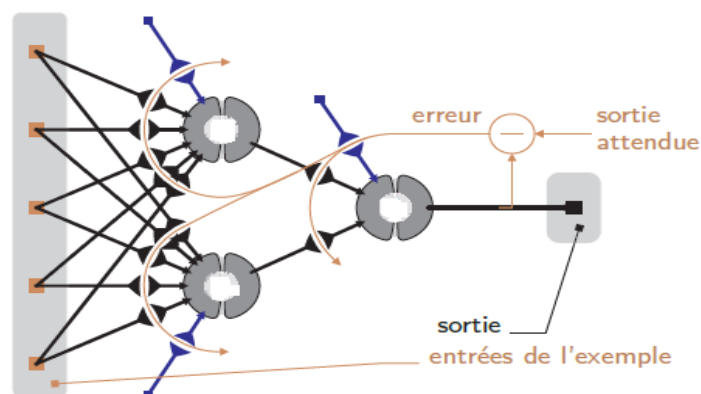
Théoriquement, un MLP doté d'une couche cachée est capable de résoudre tous les problèmes de classification à condition que le nombre d'exemples nécessaires à son apprentissage soit suffisant.

### 3.2.2. L'algorithme d'apprentissage :

L'algorithme d'apprentissage est une activité essentielle à la base même de toute évolution. C'est un processus calculatoire qui doit être capable d'amener à une certaine prédiction et à une certaine généralisation, son but est de découvrir les règles qui gouvernent et régissent des formes. L'apprentissage des réseaux de neurones est la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée [ZIGHED\_2000].

Les réseaux de neurones utilisant un apprentissage par rétro-propagation [PINEDA\_1987], [RUMEL\_1986] sont constitués de plusieurs couches (multicouches) qui comportent chacune plusieurs neurones. Ces derniers sont reliés entre eux par des matrices de poids qui caractérisent le réseau.

La rétro-propagation est un algorithme basé sur l'apprentissage supervisé. Elle procède à l'adaptation des poids neurone par neurone en commençant par la couche de sortie afin de minimiser l'erreur entre la sortie désirée et la sortie obtenue (Figure 3.3).



**Figure 3.3. Apprentissage d'un perceptron multicouche par rétro-propagation du gradient.**

Dans notre travail, nous avons utilisé un apprentissage supervisé avec le perceptron multicouche en adoptant l'algorithme de rétro-propagation par la descente de gradient.

### 3.2.3. La base de données :

Dans cette étude, nous avons utilisé les enregistrements ECG (Electrocardiogramme) de la base de données internationale MIT-BIH (*Massachusetts Institute of Technology/ Beth Israel Hospital*) [MIT\_2000]. Cette base contient 48 enregistrements de 30 minutes chacun provenant d'un moniteur cardiaque Holter. Ces enregistrements sont divisés en deux groupes, le 1<sup>er</sup> groupe contient 23 enregistrements numérotés entre 100 et 124, il est prévu pour servir d'échantillon représentatif de différentes formes d'ondes, tandis que le 2<sup>ème</sup> groupe, il est de 25 enregistrements numérotés entre 200 et 234, il est choisi pour inclure une variété de cas pathologiques. Les sujets dans la base de données sont 25 hommes âgés de 32 à 89 ans et 22 femmes âgées de 23 à 89 ans.

La base MIT-BIH nous a permis la création de deux bases de données : une pour l'apprentissage et l'autre pour le test, elles serviront pour l'entraînement et l'évaluation du classifieur. Nous avons choisi dans les deux bases des enregistrements qui contiennent déjà les battements ESVs (Extrasystoles Ventriculaires).

#### 3.2.3.1. La base d'apprentissage :

Nous avons utilisé une base de données pour la phase d'apprentissage qui contient les enregistrements représentés dans le tableau 3.1.

**Tableau 3.1. Les enregistrements utilisés pour la conception de la base d'apprentissage.**

N°	Record	Sexe	Age	N	V
<b>Pour l'apprentissage</b>					
1	105	Female	73	2526	41
2	114	Female	72	1820	43
3	200	Male	64	1743	826
4	201	Male	68	1625	198
5	202	Male	68	2061	19
6	203	Male	43	2529	444
7	205	Male	59	2571	71
8	210	Male	89	2423	194
9	215	Male	81	3195	164
10	217	Male	65	244	162
11	219	Male	Age not recorded	2082	64
12	223	Male	73	2029	473
13	228	Female	80	1688	362
14	233	Male	57	2230	831

Avec : N est le nombre de battements normaux.

V est le nombre de battements des extrasystoles ventriculaires ESV.



### 3.2.3.2. La base de test :

Nous avons sélectionné d'autres enregistrements pour la phase de test du classifieur, ils sont représentés dans le tableau 3.2.

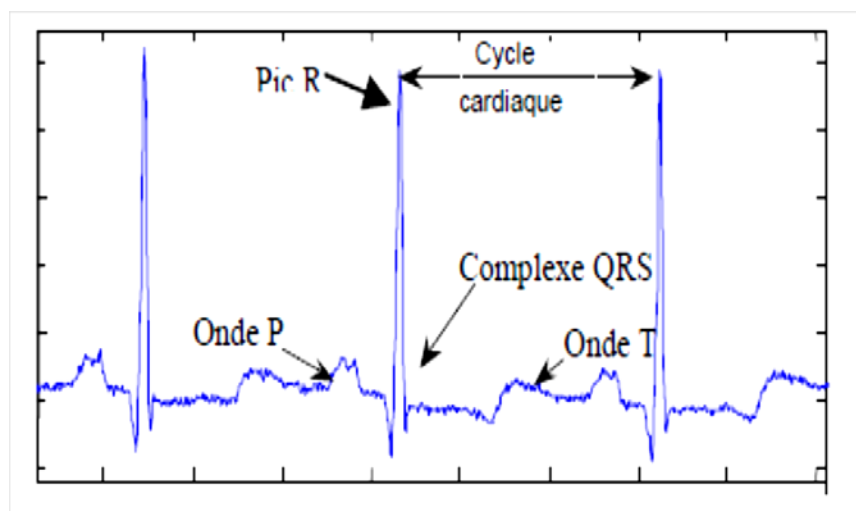
**Tableau 3.2. Les enregistrements utilisés pour la phase de test.**

N°	Record	Sexe	Age	N	V
<b>Pour le test</b>					
<b>1</b>	106	Female	24	1507	520
<b>2</b>	116	Male	68	2302	109
<b>3</b>	119	Female	51	1543	444
<b>4</b>	208	Female	23	1586	992
<b>5</b>	213	Male	61	2641	220
<b>6</b>	221	Male	83	2031	396

### 3.2.4. L'arythmie ciblée :

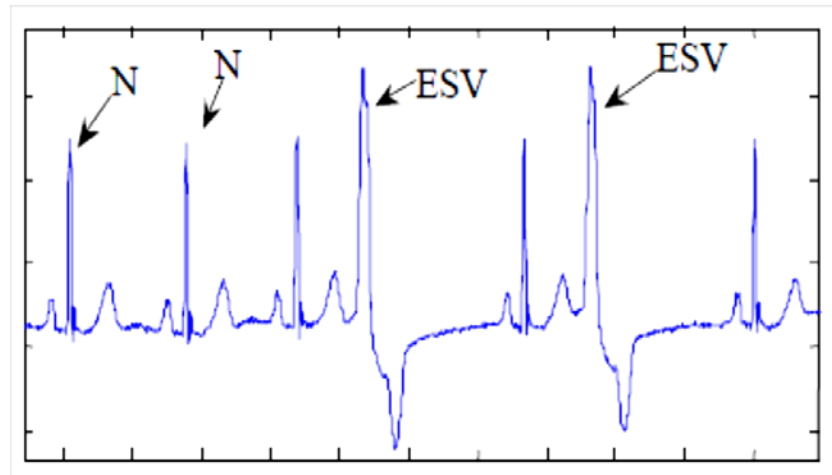
Le cœur assure la vie : c'est lui qui alimente les différents organes du corps humain en pompant le sang et en le refoulant dans le réseau des artères. Le battement est dû à une onde électrique parcourant le cœur, qui a son passage provoque la contraction des cellules musculaires cardiaque. Une arythmie correspond à une irrégularité du parcours de cette onde résultant d'un dysfonctionnement cardiaque [RENE\_2006].

La classification de différents types de cardiopathies à partir de l'électrocardiogramme ECG représente une importance majeure dans le diagnostic des dysfonctionnements cardiaques (Figure 3.4). Le choix des paramètres qui constituent le vecteur d'entrée du classifieur neuronal est lié à l'arythmie ciblée.



**Figure 3.4. ECG d'un sujet sain [CARRE\_1984].**

Dans notre travail, nous avons choisi l'extrasystole ventriculaire ESV à partir d'enregistrements d'ECG de la base d'arythmies MIT\_BIH (Figure 3.5). Cette arythmie est caractérisée par un complexe QRS prématuré et large et une absence de l'onde P.



**Figure 3.5. ECG d'un sujet qui présente des extrasystoles ventriculaires [CARRE\_1984].**

Nous avons gardé les descripteurs suivants pour caractériser un battement cardiaque :

- RRp : la distance entre l'onde R actuelle et l'onde R précédente.
- RRs : la distance entre l'onde R actuelle et l'onde R suivante.
- RRs/RRp : le rapport entre la distance RR suivante sur la précédente.
- Peak : amplitude de peak de l'onde R.
- QRS : la durée du complexe QRS.
- PP : amplitude (peak to peak) de l'onde R jusqu'à l'onde S.
- Energie : énergie moyenne du complexe QRS au carré.

Ces descripteurs sont extraits en utilisant un logiciel de mesure des paramètres [CHIKH\_2004] élaboré dans le laboratoire LTSI de Rennes-France.

Dans le cadre de cette thèse, nous avons réalisé un classifieur neuronal à une seule sortie (un neurone dans la couche de sortie). Nous avons cité deux classes :

- La classe V : est la classe positive qui représente les battements des extrasystoles ventriculaires ESV.
- La classe N : est la classe négative qui représente les battements normaux.

### 3.2.5. La normalisation des données :

Les données d'apprentissage et de test sont structurées sous forme de matrices. Le nombre de lignes représente le nombre de battements  $T(i)$  (les sorties) et le nombre de colonnes représente le nombre de caractéristiques  $P(i)$  (les entrées) quantifiant chaque battement. Chaque ligne représente donc un vecteur caractérisant un battement.

Afin d'améliorer les performances des réseaux neuronaux multicouches, il est préférable de normaliser les données d'entrée et de sortie dans l'intervalle  $[-1, +1]$ .

### 3.2.6. Apprentissage du classifieur neuronal:

Pour notre travail, nous avons utilisé la boîte à outils « Matlab Réseau de Neurones » (*Neural Network Toolbox Nntool*) [NNT\_2016], il s'agit d'une structure hiérarchique

(certains membres de la structure de base sont eux-mêmes des structures) relativement complexes.

### 3.2.6.1. L'étape d'apprentissage :

Lors de la conception de notre classifieur, nous avons ajusté un certain nombre de paramètres comme le nombre d'itérations, le seuil d'erreur, ...etc. (Figure 3.6).

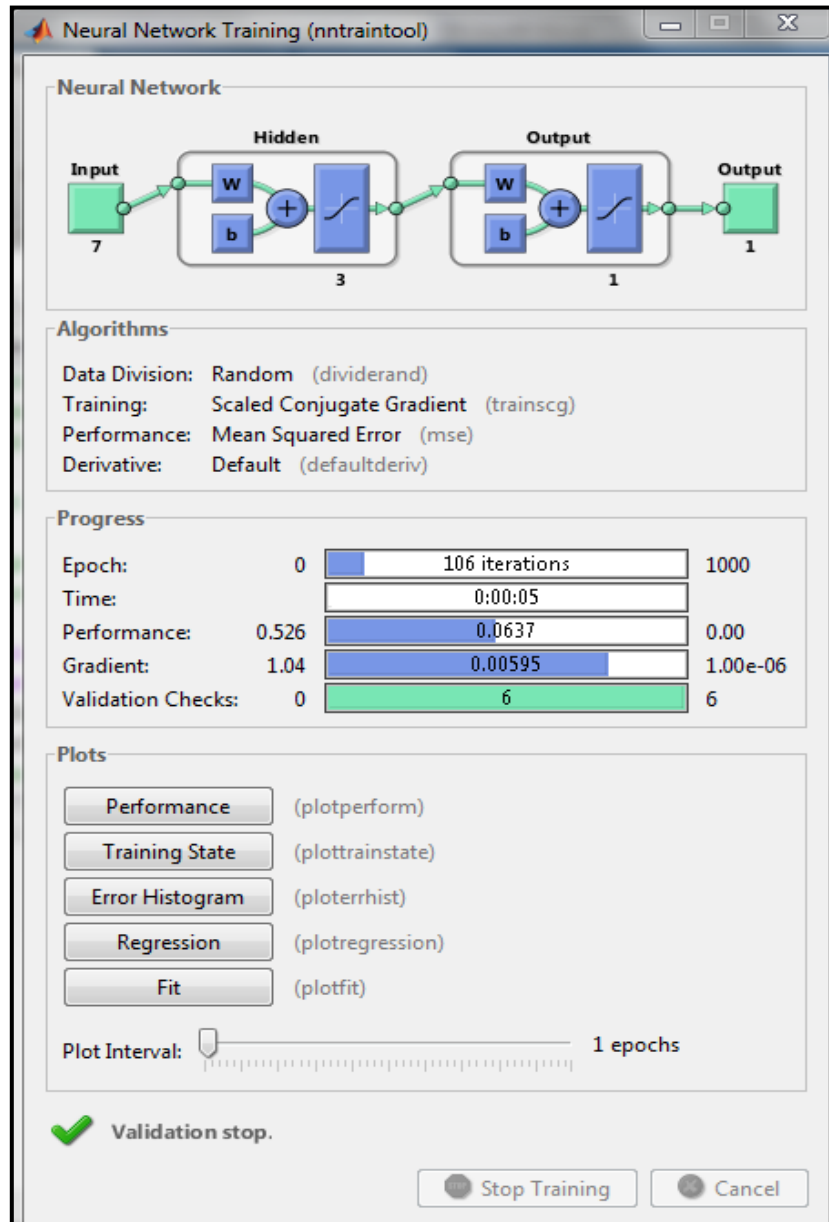
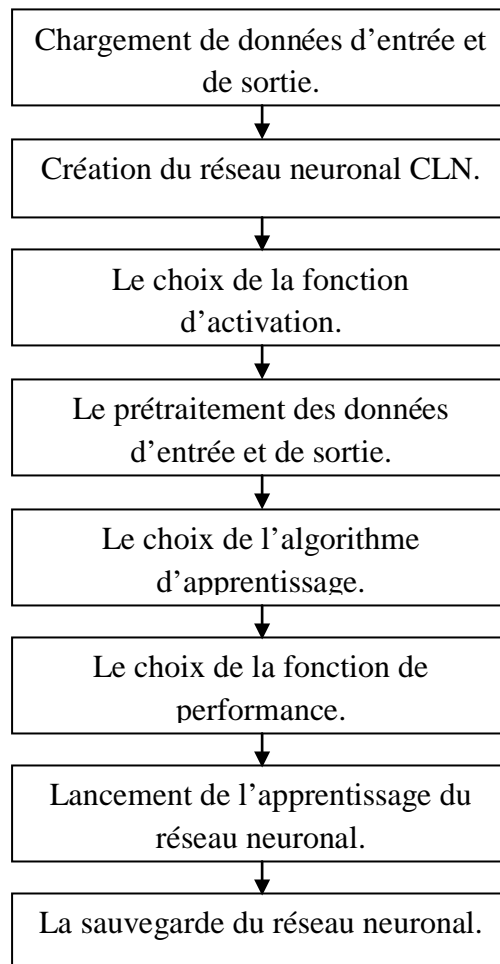


Figure 3.6. Visualisation du réseau réalisé sur la boîte à outils.

L'organigramme suivant présente les différentes étapes pour la création et l'apprentissage du réseau neuronal CLN.



Organigramme 3.1. Les différentes étapes pour l'apprentissage du classifieur neuronal CLN.

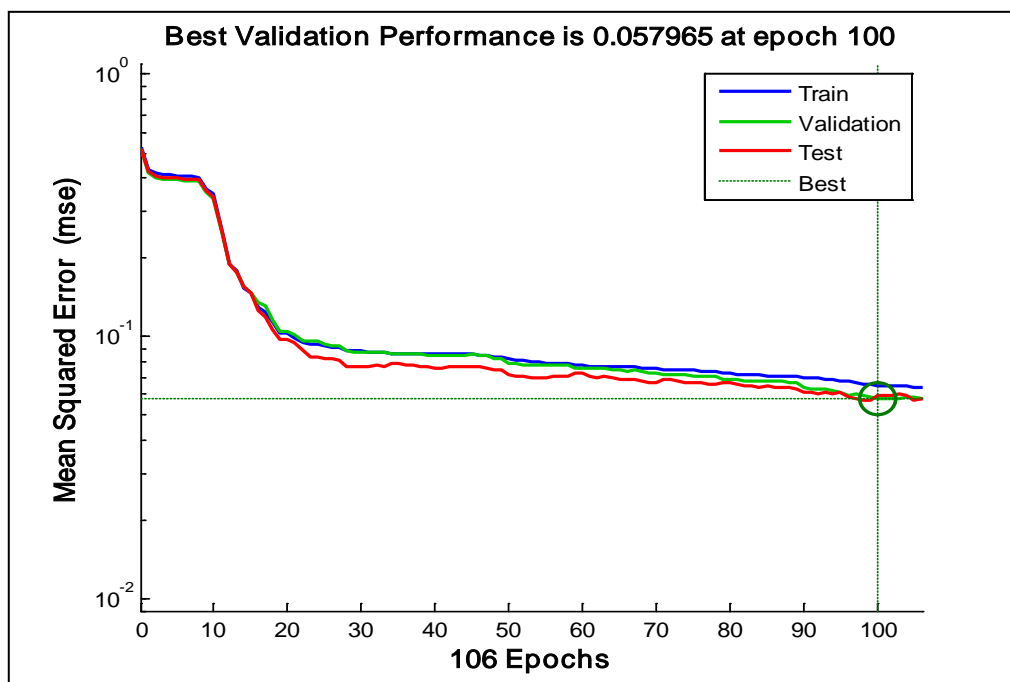


Figure 3.7. Les meilleures performances de validation.

### 3.2.6.2. L'étape de test :

Dans la figure suivante, nous remarquons le temps de calcul effectué lors d'une classification sous l'environnement Matlab (le programme est présenté dans l'annexe B).

```
>> ApvcC
nombre de cas reconnus est :
    1976

taux de reconnaissance est :
    99.9494

Le nombre de cas TP1 est :
    1535

Le nombre de cas TN1 est :
    441

Le nombre de cas FP1 est :
    0

Le nombre de cas FN1 est :
    0

Elapsed time is 9.959608 seconds.
```

Figure 3.8. Les résultats obtenus lors d'une classification.

## 3.3. La conception matérielle du classifieur neuronal CLN :

Le classifieur neuronal a été entraîné à partir de la base de données MIT-BIH. Nous avons fixé son architecture et ses paramètres après plusieurs essais. La structure finale est composée de trois couches :

- Une couche d'entrée : 7 neurones (les paramètres d'entrée).
- Une couche cachée : 3 neurones.
- Une couche de sortie : 1 neurone (la classe).

Notons que les neurones de la couche cachée et de sortie ont une fonction d'activation du type tangente hyperbolique sigmoïde.

Dans un premier temps, nous avons réalisé la conception matérielle du classifieur neuronal CLN en adoptant un principe de calcul qui s'appelle le **Semi-Parallèle** (Section 3.3.3). Ensuite, l'architecture matérielle du neurone a été ajustée en ajoutant **un détecteur de dépassement de capacité** (section 3.3.4) afin d'améliorer nos résultats et enfin, nous avons proposé un autre concept de calcul, il concerne le **Parallèle-Parallèle** en réalisant une deuxième conception matérielle du classifieur CLN (section 3.3.5) dans le but de diminuer le temps de réponse de notre classification.

### 3.3.1. Représentation des données :

Avant de commencer l'implémentation matérielle, il faut faire une normalisation des données (les valeurs des neurones d'entrée, les valeurs des poids synaptiques et du biais de chaque neurone). Pour cela, nous avons penché vers la quantification en virgule fixe pour les

données réelles dont le nombre de points et la distance entre les points sont arbitraires, mais le plus souvent nous choisissons un ensemble symétrique par rapport à l'origine.

Évidemment, le nombre de valeurs représentable détermine le nombre de bits qui sont nécessaires pour leur représentation [BEUCHAT\_2001]. La façon la plus naturelle est de réserver un bit de signe qui est le bit de poids fort ou *MSB (Most Significant Bit)* avec le symbole logique 1 pour le nombre négatif et le symbole 0 pour le nombre positif, le reste des bits pour représenter le nombre comme fraction binaire (Figure 3.9).



Figure 3.9. Représentation d'un nombre en binaire.

Pour les données signées, nous avons choisi la représentation par complément à deux, elle consiste à remplacer les nombres négatifs et les écrire comme fraction binaire. Cette représentation est très répandue parce qu'elle présente des avantages pour les opérations arithmétiques (Figure 3.10).

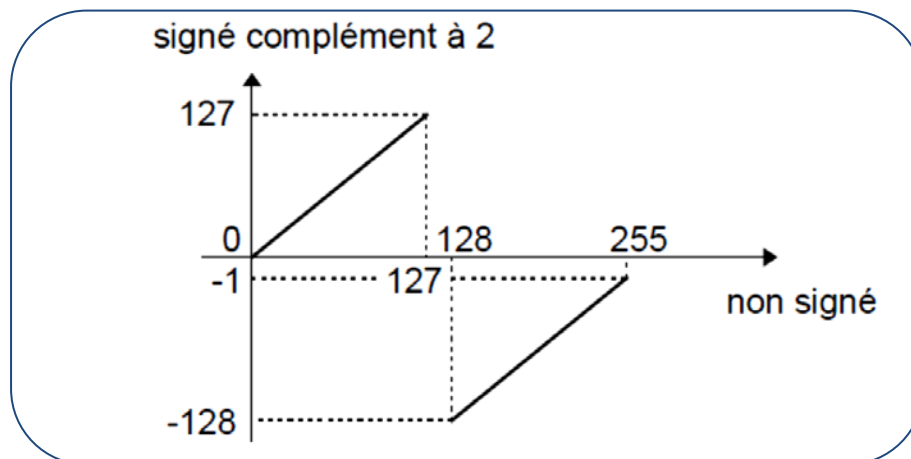


Figure 3.10. Représentation des nombres signés et non signés sur 8 bits.

Dans notre architecture matérielle, nous avons choisi la représentation des données sur 8 bits dans un espace entre -1 et +1 avec une précision de 1/128, nous présentons quelques nombres dans le tableau suivant:

**Tableau 3.3. Représentation des données.**

<b>Nombres</b>	<b>Représentation</b>
<b>-1</b>	10000000
<b>-0,5</b>	11000000
<b>-0,25</b>	11100000
<b>0</b>	00000000
<b>0,0078125</b>	00000001
<b>0,046875</b>	00000110
<b>0,25</b>	01100000
<b>0,5</b>	01000000
<b>0,9921875</b>	01111111

### **3.3.2. Le principe de calcul :**

Dans notre thèse, nous avons abordé deux principes de calcul différents Semi-Parallèle (SP) et Parallèle-Parallèle (PP) afin d'avoir un compromis entre les performances relatives au temps de calcul et les besoins en ressources matérielles dans le circuit FPGA.

#### **3.3.2.1. Le Semi-Parallèle (SP) :**

Le principe de calcul 'Semi-Parallèle' (SP) consiste à transmettre l'information séquentiellement d'une couche vers l'autre, en effectuant un calcul parallèle dans la même couche du classifieur matériel (CLNH-SP). En utilisant ce principe, l'occupation de ressources dans le FPGA est minimale au détriment du temps d'exécution.

#### **3.3.2.2. Le Parallèle-Parallèle (PP) :**

L'application de ce principe (PP) permet à transmettre l'information d'une couche vers l'autre en parallèle, en effectuant un calcul parallèle dans la même couche du classifieur matériel (CLNH-PP). Ce type d'implantation peut vite s'avérer comme un consommateur de ressources, en fournissant un temps d'exécution rapide.

### **3.3.3. L'architecture matérielle 'CLNH-SP' du classifieur neuronal :**

La réalisation matérielle du classifieur 'CLNH-SP' est basée sur deux éléments principaux : le réseau neuronal et le séquenceur. Ce dernier est une sorte de machine d'état qui enchaîne les étapes de calcul nécessaires pour la classification des paramètres d'entrée, et le réseau neuronal contient un élément de base 'Neurone\_NC', qui sera détaillé en premier lieu [BABA\_10/2014].

#### **3.3.3.1. La conception matérielle d'un neurone non complet :**

Afin d'optimiser au maximum notre architecture, il est important de bien comprendre les opérations mathématiques du réseau de neurones. Pour notre classifieur CLN, chaque neurone formel réalise deux phases distinctes (Figure 3.11) :

- La première phase, appelée activation représente le calcul de la somme pondérée des entrées du neurone.
- La deuxième phase concerne une fonction de transfert qui évalue la sortie du neurone.

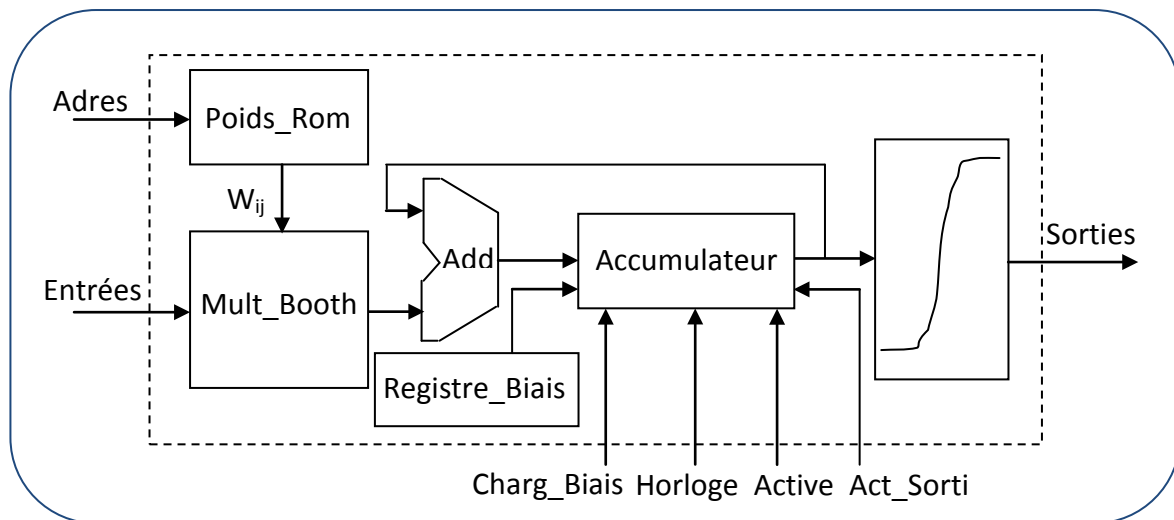


Figure 3.11. Architecture matérielle d'un seul neurone.

Les sorties des neurones de la couche d'entrée arrivent aux entrées des neurones de la couche cachée successivement (les paramètres des arythmies cardiaques sont détectés séquentiellement), dans cette couche, les calculs s'effectuent au niveau de chaque neurone en parallèle puis leurs réponses sont envoyées à la couche de sortie en série, ce qui nous amène à appeler ce principe de calcul par **Semi-Parallèle**, donc nous pouvons utiliser qu'un seul bloc de la fonction d'activation pour les neurones de la couche cachée au lieu de trois blocs. Pour cela, nous allons ôter le bloc de la fonction d'activation de l'architecture du neurone et l'utiliser comme un bloc individuel après chaque couche (voir section 3.3.3.2.). Cela permet la minimisation de nombre de portes logiques utilisées dans le FPGA ce qui conduit directement à une minimisation dans le temps de réponse.

Donc l'architecture matérielle d'un neurone sera composée seulement d'une ROM, d'un multiplieur, d'un registre et d'un accumulateur. L'ensemble de ces blocs réalise le bloc **Neurone\_NC** (neurone non complet).

Le processus est synchronisé sur une horloge (Horloge) dont le nombre de cycles pour qu'un Neurone\_NC termine ses calculs, égale au nombre de connexions de ce Neurone\_NC à la couche précédente.

#### ✚ La mémoire ROM :

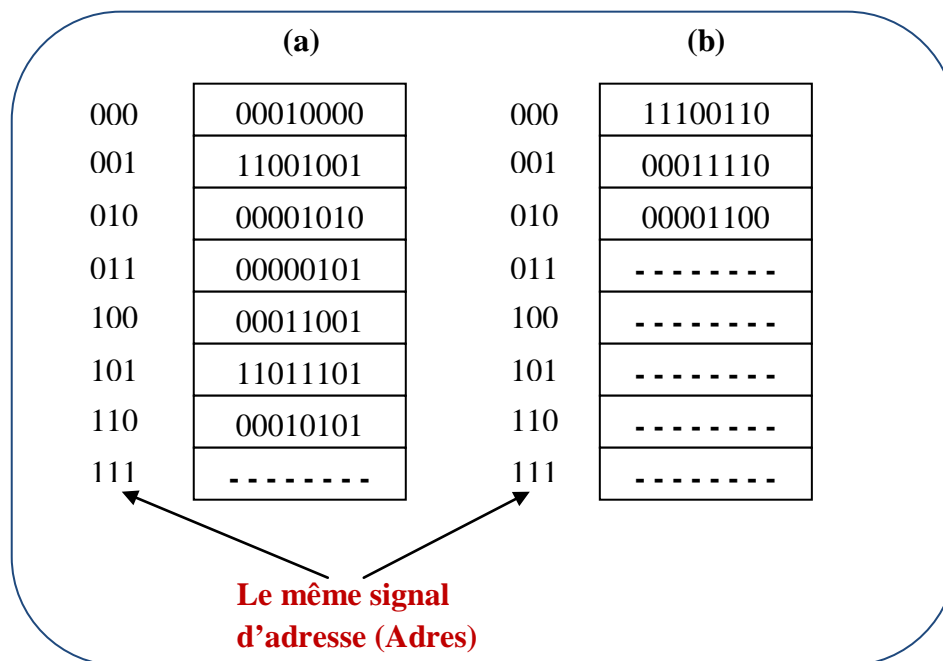
Nous avons utilisé une mémoire du type ROM (*Read Only Memory*) pour stocker les valeurs des poids synaptiques correspondantes à ce neurone (Poids\_Rom).

Chaque neurone de la couche cachée doit avoir sept valeurs des poids synaptiques donc la taille de la ROM sera de  $2^3$  lignes. Le neurone de la couche de sortie a trois valeurs de poids



synaptiques donc sa ROM doit avoir  $2^2$  lignes mais afin d'utiliser le même signal d'adresse (Adres) pour la lecture de ces valeurs, nous avons généralisé la taille de tous les mémoires dans le classifieur neuronal à  $2^3$  lignes.

La figure 3.12 présente la gestion des mémoires ROM dans les neurones des deux couches (la couche cachée et la couche de sortie), nous avons utilisé dans le langage VHDL (*Very high speed integrated circuits Hardware Description Language*) le paquetage *ieee.std\_logic\_1164* qui définit le type *std\_logic\_vector* de la librairie IEEE [WEBER\_2007], ce type comporte le caractère « ----- » (*Don't care*) qui est mis dans les lignes non utilisées afin d'avoir des mémoires remplies correctement.



**Figure 3.12. La gestion des mémoires ROM dans le neurone.**  
(a) Rom d'un neurone de la couche cachée. (b) Rom du neurone de la couche de sortie.

Ce mode de lecture dans les différentes ROM nous a permis de généraliser l'architecture du Neurone\_NC dans le classifieur neuronal CLNH-SP.

#### Le multiplieur :

Les neurones de la couche précédente sont liés à ce neurone en série et ils sont multipliés par leurs pondérations correspondantes en utilisant un multiplieur. Ce dernier est basé sur l'algorithme de BOOTH qui permet de multiplier deux nombres binaires signés représentés en complément à deux.

Booth a découvert son algorithme en cherchant une rapidité maximale, il était persuadé qu'un décalage était plus rapide qu'une addition [CHANDEL\_2013]. Son principe est simple,

soit : **a** le multiplicateur et **b** le multiplicande. En s'intéressant aux bits du multiplicateur dans l'algorithme de Booth, nous obtenons le tableau 3.4.

**Tableau 3.4. Le principe de l'algorithme de BOOTH.**

<b>a<sub>1</sub></b>	<b>a<sub>0</sub></b>	<b>Opérations</b>
<b>0</b>	<b>0</b>	Un décalage à droite.
<b>0</b>	<b>1</b>	Ajouter b puis un décalage à droite.
<b>1</b>	<b>0</b>	Soustraire b suivi d'un décalage à droite.
<b>1</b>	<b>1</b>	Un décalage à droite.

La multiplication de deux nombres de 8 bits chacun donne un produit de 16 bits, et puisque la représentation des données est codée sur 8 bits, nous avons besoin d'une remise en forme pour les sorties. Pour cela, nous utilisons une simple troncature sur les vecteurs de 16 bits de façon à préserver le bit de signe.

#### ✚ Le registre :

Le registre est un élément de stockage qui permet la mémorisation de n bits en parallèle. Il est constitué de n bascules en parallèle, mémorisant chacune un bit. Les bascules employées sont souvent du type D (*Flip Flop*).

Nous avons utilisé un registre (Registre\_Biais) de 8 bits, qui contient la valeur du biais, ce dernier sera chargé dans l'accumulateur au démarrage en activant le signal (Charg\_Biais) [BABA\_06/2014].

#### ✚ L'accumulateur :

L'accumulateur est un registre qui contient au démarrage du système la valeur du biais, puis il va accumuler les entrées pondérées par leurs poids synaptiques à chaque cycle d'horloge par un additionneur (Add). La taille de l'accumulateur est de 8 bits.

En s'appuyant sur la représentation des données en compléments à 2 sur 8 bits, nous pouvons avoir un dépassement de capacité (*Overflow*) qui intervient si le résultat d'une opération arithmétique n'est pas représentable dans le système. Pour la multiplication, nous n'aurons pas des erreurs car les données sont représentées dans l'intervalle [-1, +1], mais nous craignons que des débordements de calcul se produisent au cours de l'accumulation des entrées pondérées. Pour éviter cela, nous avons choisi de diviser le résultat sur le nombre d'accumulations effectué pendant le calcul, par exemple, un neurone de la couche cachée exécute 8 accumulations (les sept accumulations des entrées pondérées plus le biais).

#### 3.3.3.2. Le réseau neuronal :

Dans un deuxième lieu, Nous allons détailler le réseau neuronal (Figure 3.13) qui est composé de deux couches (cachée et de sortie), deux blocs de la fonction d'activation et un comparateur.

En activant le signal (Active1), les valeurs des sept paramètres d'entrée arrivent séquentiellement à l'entrée de la couche cachée d'où ses trois Neurone\_NC sont reliés. Ces derniers calculent leurs entrées pondérées en même temps (en parallèle) et procèdent trois valeurs en sortie qui doit passer successivement à la fonction d'activation (Fonct\_Tansig) en activant l'un des signaux de sortie de chaque Neurone\_NC de la couche cachée (Act\_Sorti1, Act\_Sorti2 et Act\_Sorti3).

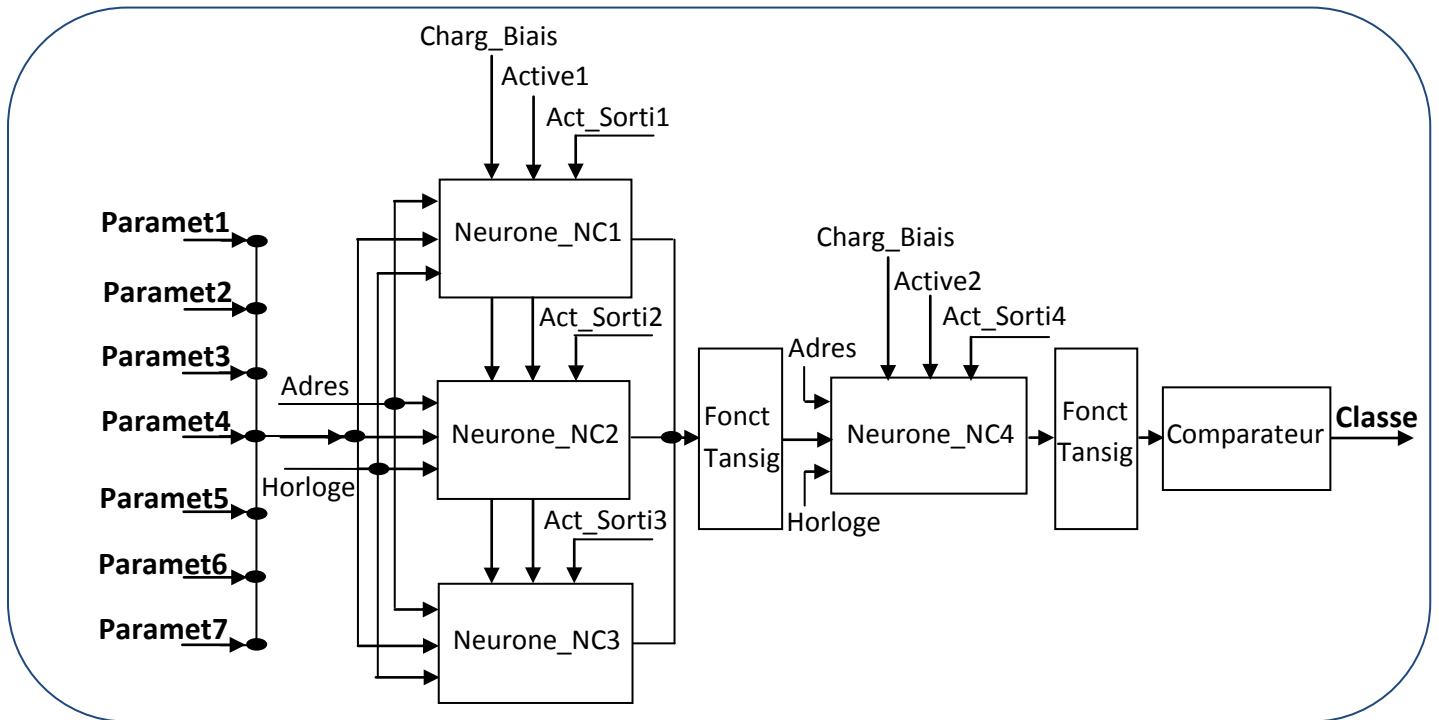


Figure 3.13. Architecture matérielle du réseau neuronal.

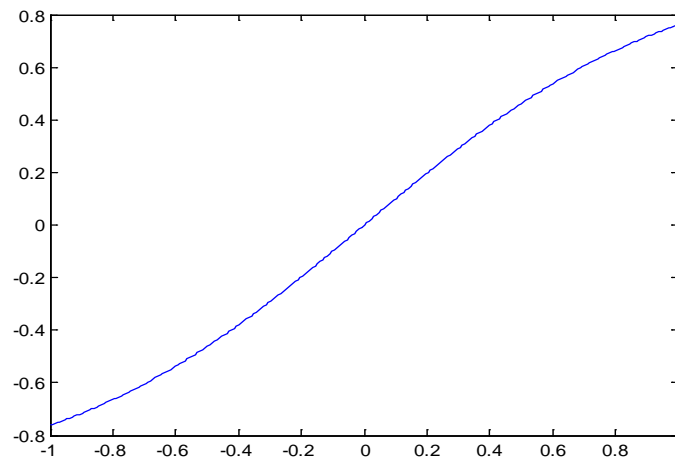
Après le passage des réponses de la couche cachée à travers la fonction d'activation, le signal (Active1) est désactivé et les réponses arrivent au Neurone\_NC de la couche de sortie en série en activant le signal (Active2). Si le signal (Act\_Sorti4) est activé, la sortie de ce Neurone\_NC est transmise à la deuxième fonction d'activation, puis elle traverse le bloc comparateur.

**La fonction d'activation :**

Une des parties les plus importantes d'un neurone est sa fonction d'activation, et l'implémentation de cette fonction est une tâche difficile que rencontrent les concepteurs surtout pour les fonctions non linéaires comme la fonction Tangente Hyperbolique Sigmoidale « Tansig » (utilisée dans notre classifieur), elle est présentée par l'équation suivante :

$$F(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (3.3)$$

Pour l'approximation non linéaire de notre fonction Tansig (Figure 3.14), nous avons choisi l'utilisation d'un générateur de fonctions appelé LUT (Look Up Table) qui se trouve dans les blocs logiques du FPGA. Cette LUT est une partie mémoire dont nous pouvons stocker les valeurs des échantillons uniformes prélevés de la fonction Tansig dans l'intervalle [-1, +1] afin d'obtenir une approximation plus précise.



**Figure 3.14. La fonction Tangente Hyperbolique Sigmoide.**

La mise en œuvre directe des fonctions non linéaires (la fonction Tansig) en utilisant les LUT est très coûteuse, car c'est des mémoires qui nécessitent beaucoup de ressources logiques du FPGA et comme nous utilisons un principe de calcul Semi-Parallèle (le calcul se fait en parallèle dans la même couche et l'information passe d'une couche à l'autre en série), nous avons utilisé un seul bloc de la fonction d'activation pour tous les Neurone\_NC de la même couche. Cela permet une minimisation du nombre de portes logiques utilisées dans le FPGA ce qui implique directement une minimisation du temps de réponse.

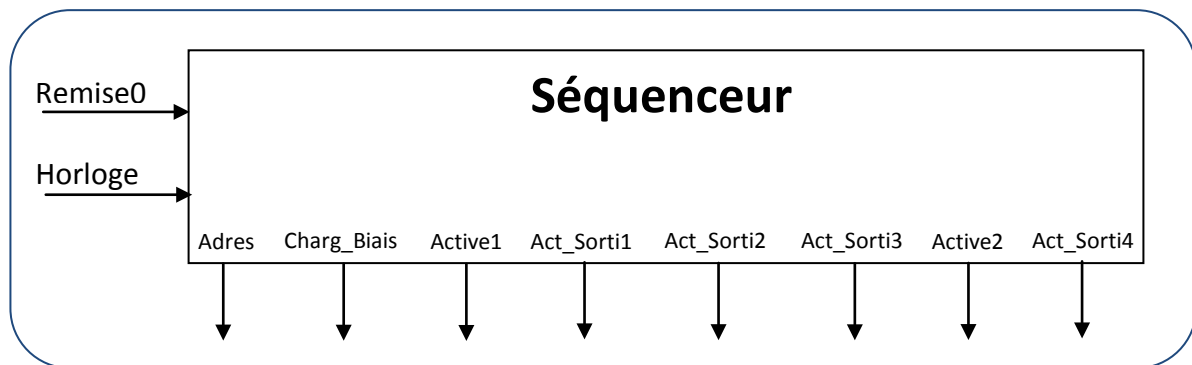
#### **✚ Le comparateur :**

Le résultat du calcul dans le classifieur neuronal arrive au bloc comparateur sous la forme d'un vecteur de 8 bits dont le bit le plus significatif (MSB) est le bit de signe. Le bloc a pour but de normaliser cette valeur calculée vers 0 ou 1 :

- Si la valeur calculée est négative (MSB=1), elle est arrondie vers la classe 0 (N : cas normal).
- Si la valeur calculée est positive (MSB=0), elle est arrondie vers la classe 1 (V : cas pathologique).

#### **3.3.3.3. Le séquenceur :**

Le séquenceur (Figure 3.15) est un automate distribuant selon un chronogramme précis (Figure 3.16), des signaux de commande aux divers blocs du réseau neuronal participant à l'exécution des instructions.



**Figure 3.15. Bloc du Séquenceur.**

Ce bloc contrôle l'ensemble des opérations du réseau :

- L'enchaînement des adresses des ROM qui contiennent les valeurs des poids synaptiques.
- Le chargement des valeurs des biais dans les accumulateurs de chaque Neurone\_NC.
- L'activation de la couche cachée après 7 cycles d'horloge (le nombre de neurones d'entrée).
- L'activation de la couche de sortie après 3 cycles d'horloge (le nombre de Neurone\_NC de la couche cachée).
- L'activation des signaux qui permettent le passage des sorties successivement couche après couche.

Notons que tous les signaux de commande sont actifs à l'état haut:

- ✓ Remise0 : Le signal de remise à zéro du classifieur neuronal.
- ✓ Horloge : C'est l'horloge qui existe dans le circuit reconfigurable FPGA.
- ✓ Charg\_Biais : A l'activation de ce signal, la valeur du biais est chargée dans l'accumulateur.
- ✓ Adres(0) : Le premier bit du signal d'adresse de la ROM.
- ✓ Adres(1) : Le deuxième bit du signal d'adresse de la ROM.
- ✓ Adres(2) : Le troisième bit du signal d'adresse de la ROM.
- ✓ Active1 : Le signal d'activation de la couche cachée.
- ✓ Act\_Sorti1 : À l'activation de ce signal, la sortie du Neurone\_NC1 passe à la couche de sortie.
- ✓ Act\_Sorti2 : À l'activation de ce signal, la sortie du Neurone\_NC2 passe à la couche de sortie.
- ✓ Act\_Sorti3 : À l'activation de ce signal, la sortie du Neurone\_NC3 passe à la couche de sortie.
- ✓ Active2 : Le signal d'activation de la couche de sortie.
- ✓ Act\_Sorti4 : Ce signal permet d'avoir la classe.

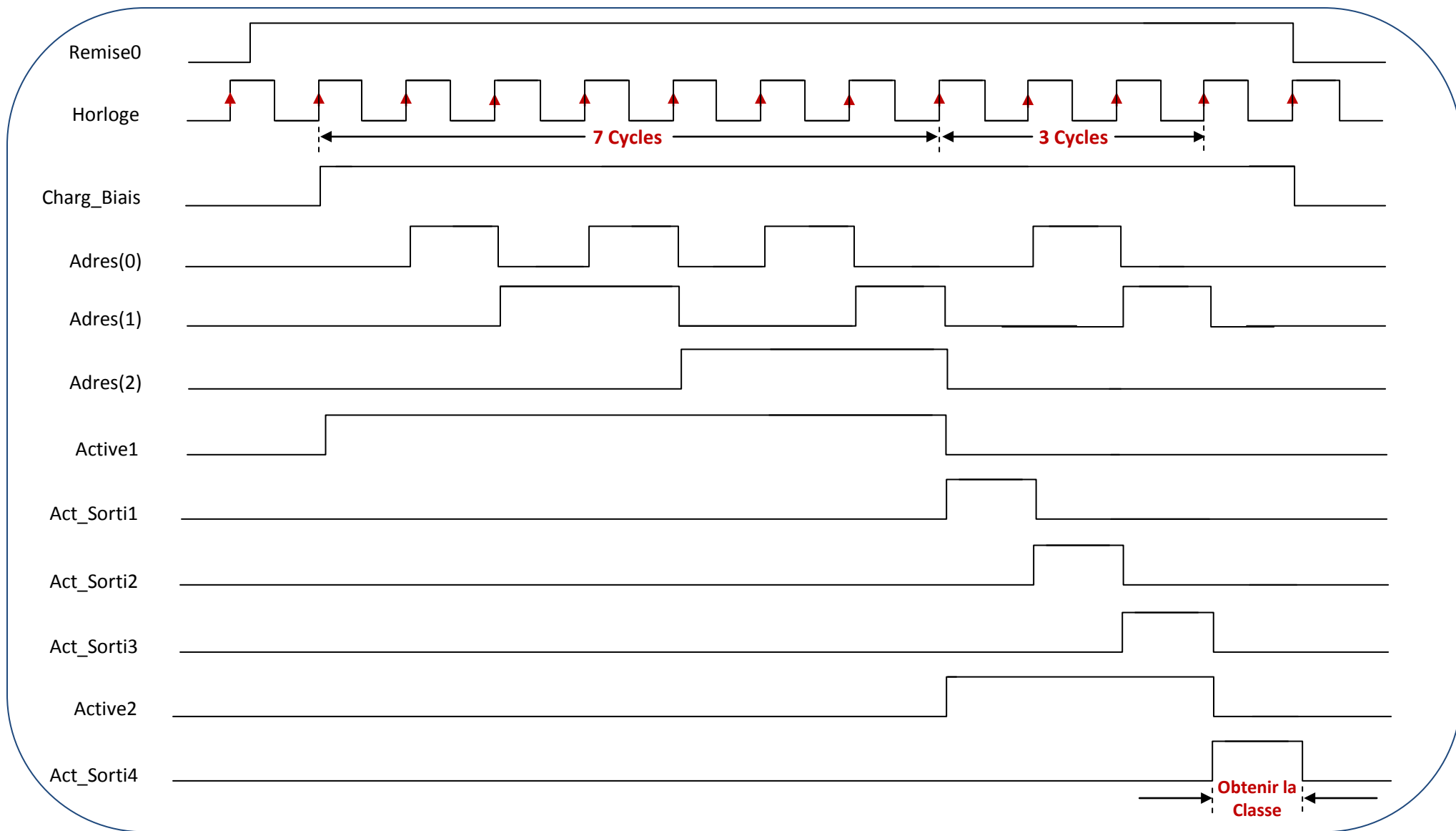


Figure 3.16. La gestion des signaux de commande dans le séquenceur.

**3.3.3.4. Remarques :**

Les différentes architectures sont synchronisées et permettent un meilleur rendement. Il s'agit donc d'avoir un bon contrôle des opérations pour rendre l'architecture efficace dans son ensemble.

Afin d'écartier le risque des dépassements de capacité dans les calculs, nous avons divisé le résultat des accumulations des entrées pondérées sur le nombre d'opérations effectuées au niveau de chaque Neurone\_NC, mais cela nous a conduits à la perte d'informations, ce qui nous a obligé à ajouter un détecteur de dépassement dans la structure du Neurone\_NC pour s'assurer qu'il y a un débordement dans le calcul avant de diviser le résultat des opérations.

**3.3.4. La détection de dépassement de capacité :**

Dans notre architecture matérielle, la représentation des données est codée sur 8 bits dans un espace qui varie entre -1 et +1. Le calcul effectué au niveau de chaque Neurone\_NC*i* est :

$$b_i + W_{i1} * P_1 + W_{i2} * P_2 + \dots + W_{ij} * P_j = a_i \quad (3.4)$$

Avec :

$b_i$  et  $W_{ij}$ : les valeurs du biais et des poids synaptiques respectives pour un Neurone\_NC*i* et qui appartient à l'intervalle [-1, +1].

$P_j$  : les valeurs des neurones d'entrée, elles sont normalisées avant la phase d'apprentissage.

Dans la 1<sup>ère</sup> couche cachée, chaque Neurone\_NC effectue 7 multiplications et 8 additions, pour la multiplication des paramètres d'entrée par leurs poids synaptiques, le résultat est sans dépassement car chacun d'eux appartient à l'intervalle [-1, +1], mais l'addition de ces entrées pondérées avec le biais peut générer un débordement de calcul. Ce qui nous a menés à changer la structure du Neurone\_NC en ajoutant un détecteur de dépassement [BABA\_2016].

Le résultat d'une addition est supposé être dans la plage de la dynamique choisie pour représenter les nombres. Si notre représentation des données est codée sur 8 bits en complément à 2, alors le résultat doit être dans l'intervalle -128 à +127. Le débordement de calcul (*arithmetic overflow*) aura lieu si le résultat de l'opération ne se trouve pas dans cette plage.

Pour s'assurer du bon fonctionnement de l'additionneur et pour avoir un résultat correct, il faut être capable de détecter ce débordement. Pour cela, nous avons ajouté un test pour les bits les plus significatifs (MSB) des deux nombres à sommer et le résultat de cette addition.

Si nous avons  $A+B=S$ , chaque terme est codé sur 8 bits et représenté en complément à 2, nous distinguons le dépassement de capacité (Ovf) dans le tableau suivant :

Tableau 3.5. Le débordement de calcul (*overflow*).

A(7)	B(7)	S(7)	Ovf	Explications
0	0	0	0	Cas normal
0	0	1	1	<b>L'addition de 2 nombres positifs donne un nombre négatif.</b>
0	1	0	0	Cas normaux
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	<b>La somme de 2 nombres négatifs donne un nombre positif.</b>
1	1	1	0	Cas normal

Sachant que : A(7) est le MSB du nombre A.  
 B(7) est le MSB du nombre B.  
 S(7) est le MSB de la somme de A et B.

Nous avons un débordement de calcul lorsque la somme de deux nombres positifs est un nombre négatif ou l'addition de deux nombres négatifs donne un nombre positif (tableau 3.5).

En testant notre classifieur, nous avons remarqué que le débordement est détecté dans l'addition qui donne un dépassement de capacité. Dans notre cas, nous avons 8 accumulations dans la couche cachée et 4 accumulations dans la couche de sortie afin de reconnaître la classe. Le dépassement de calcul peut être détecté dans l'une de ces accumulations et il peut disparaître dans la prochaine, donc il faut maintenir le signal Ovf à 1 jusqu'à la fin de cette classification pour indiquer que le calcul de cette classe est erroné. Pour cela, nous avons utilisé une bascule qui va afficher un 1 à la première apparition du débordement (Ovf) et le maintenir jusqu'à la fin (Depass). La sortie de la bascule sera initialisée à 0 à chaque classification (Figure 3.17).

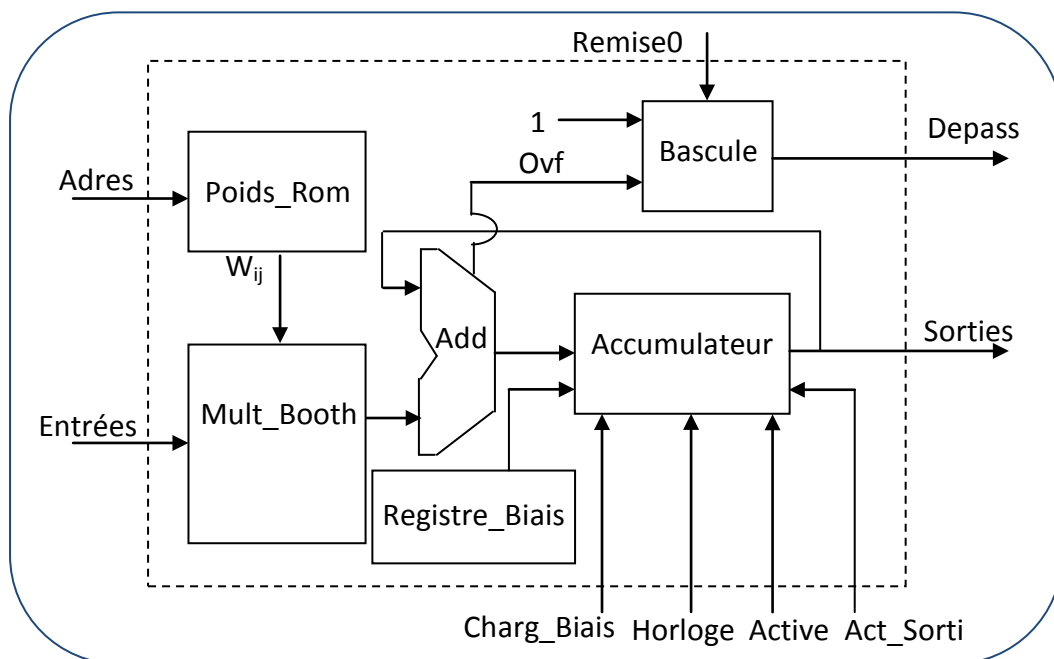
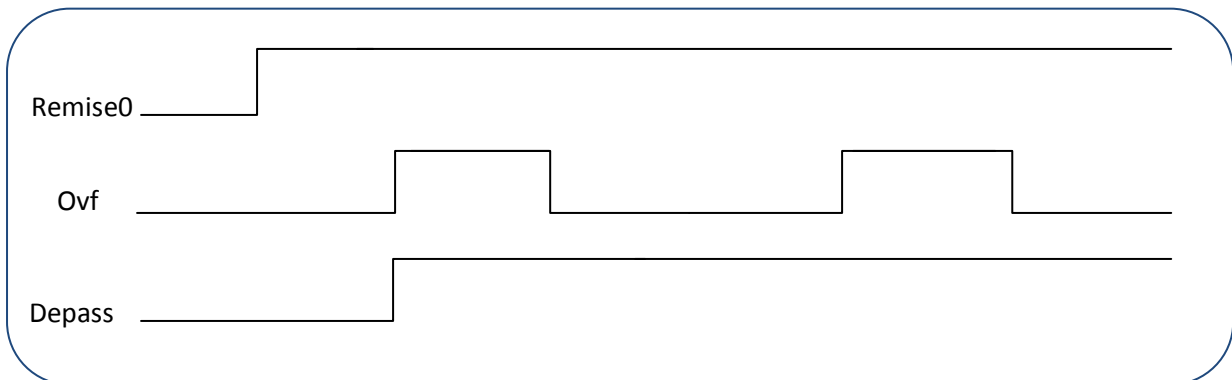


Figure 3.17. La nouvelle architecture matérielle du Neurone\_NC.



Dans la nouvelle architecture matérielle du Neurone\_NC, nous avons utilisé une bascule D synchrone munie d'une entrée dont sa valeur est « 1 », d'une entrée d'horloge connectée au signal de débordement détecté (Ovf) par l'additionneur et d'un signal de remise à zéro (Remise0) pour initialiser la bascule lors d'une nouvelle classification.

Le chronogramme de la bascule qui décrit son fonctionnement dans le bloc Neurone\_NC est illustré dans la figure 3.18.



**Figure 3.18. Le fonctionnement d'une bascule lors de la détection du dépassement de capacité.**

La minimisation du nombre de ressources logiques utilisées dans le FPGA permet de minimiser le temps de réponse d'une classification, nous n'avons utilisé qu'une seule bascule dans chaque couche (la couche cachée et la couche de sortie).

Notons que : Ovf1, Ovf2, Ovf3 et Ovf4 sont les signaux de détection des dépassements de capacité dans les Neurone\_NC1, Neurone\_NC2, Neurone\_NC3 et Neurone\_NC4 respectivement, si l'un de ces signaux est détecté alors le résultat de la classification est erroné. Pour cela, nous avons utilisé dans la couche cachée (Figure 3.19) qu'une seule bascule D pour la détection du dépassement (Depass1), le signal d'horloge de cette bascule est transmis par la sortie de la porte OU (OR) qui regroupe les trois signaux (Ovf1, Ovf2, Ovf3). Dans la couche de sortie, nous avons un seul Neurone\_NC qui nécessite une autre bascule D pour la détection du dépassement (Depass2).

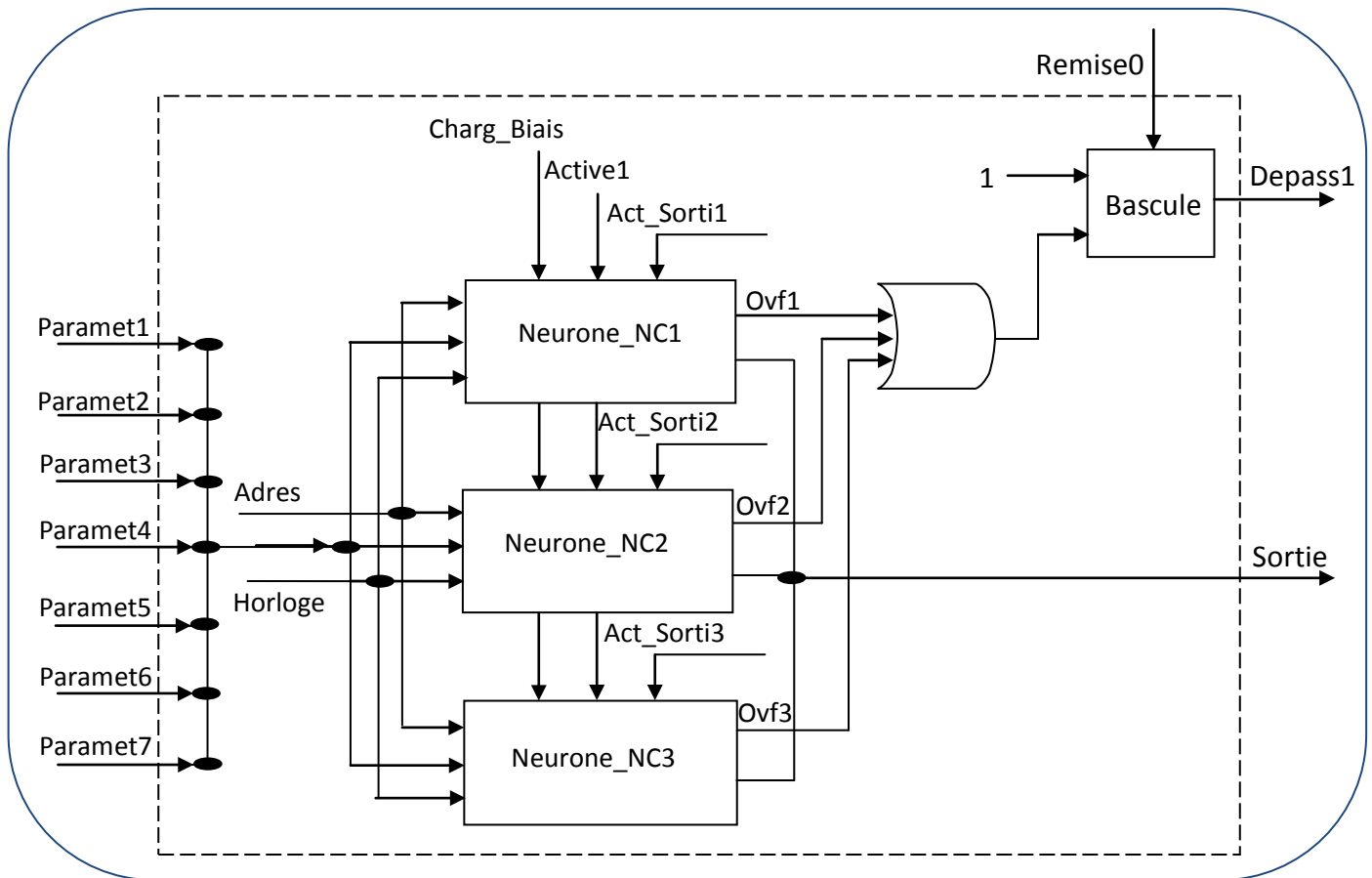


Figure 3.19. Architecture matérielle de la couche cachée du classifieur neuronal CLN.

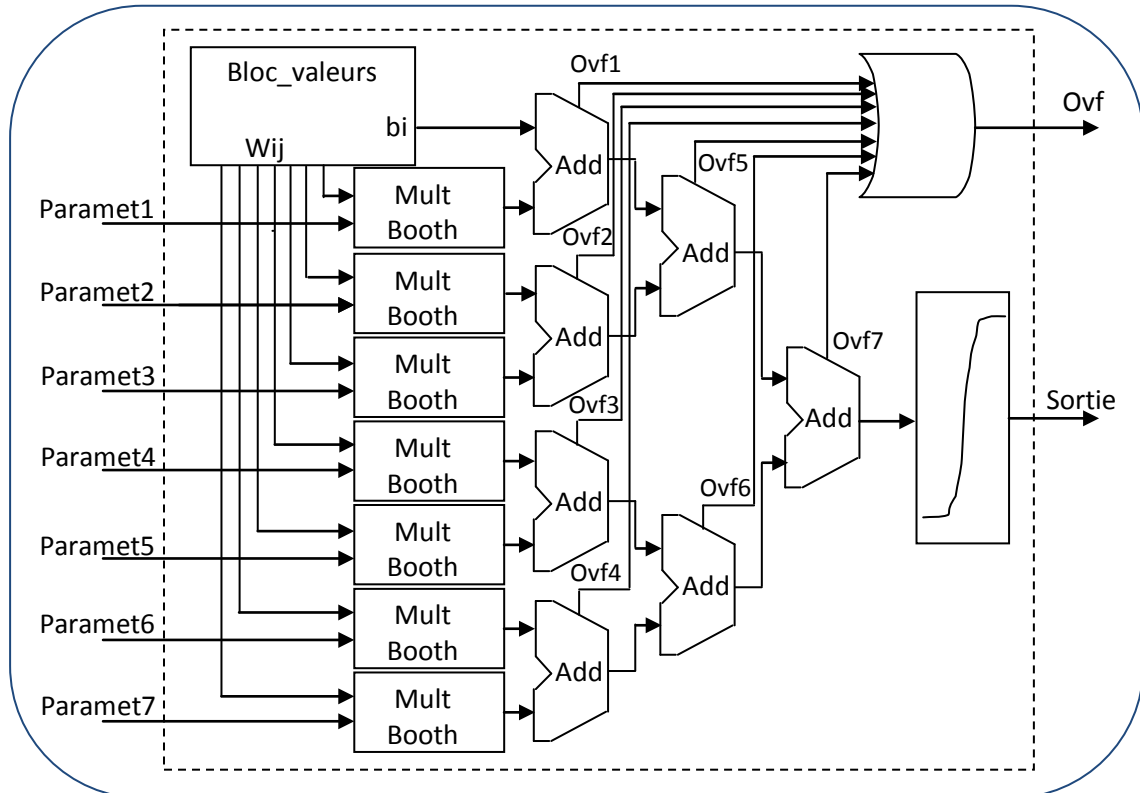
### 3.3.5. L'architecture matérielle 'CLNH-PP' du classifieur neuronal :

Dans cette partie, nous proposons un autre principe de calcul **parallèle-parallèle** dans le même classifieur neuronal afin de détecter l'extrasystole ventriculaire ESV en temps réel. Les coefficients des poids synaptiques et du biais de chaque neurone seront utilisés dans la nouvelle architecture matérielle du classifieur neuronal.

En utilisant la structure du classifieur CLN et en se basant sur le nouveau principe de calcul, les valeurs des neurones d'entrée arrivent à chaque neurone de la couche cachée en même temps, dans cette dernière le calcul s'effectue constamment en parallèle, puis leurs réponses passent à la couche de sortie en même temps.

#### 3.3.5.1. La concrétisation d'un neurone complet :

Le calcul des entrées pondérées dans un neurone s'effectue en parallèle, donc le nombre de multiplieurs et d'additionneurs dépendra du nombre des entrées à ce neurone. Normalement, les trois neurones de la couche cachée sont composés chacun de 7 multiplieurs et 7 additionneurs (Figure 3.20) et le neurone de la couche de sortie ne possède que 3 multiplieurs et 3 additionneurs.



**Figure 3.20. Architecture matérielle d'un neurone de la couche cachée.**

La conception matérielle d'un neurone est composée d'un bloc (Bloc\_Valeurs), des multiplieurs (Mult\_Booth), des additionneurs (Add) et du bloc de la fonction d'activation.

#### ✚ Le bloc des poids synaptiques et du biais :

Les valeurs des poids synaptiques et du biais de chaque neurone doivent être mises dans un bloc combinatoire (Bloc\_Valeurs) dont la lecture de ces valeurs s'effectue en même temps.

#### ✚ Le multiplieur :

Nous avons utilisé le même multiplieur déjà présenté dans la première architecture matérielle du réseau neuronal, il est basé sur l'algorithme de BOOTH. La remise en forme du résultat de cette multiplication est nécessaire car la représentation des données est codée sur 8 bits. Pour cela, nous utilisons une simple troncature sur les vecteurs de 16 bits de façon à préserver le bit de signe.

#### ✚ L'additionneur :

Après la multiplication des entrées avec leurs poids synaptiques, les résultats seront additionnés deux par deux dans un neurone sans oublier d'ajouter la valeur du biais. Nous avons utilisé l'additionneur présenté dans l'architecture matérielle du Neurone\_NC (section 3.3.4) qui peut détecter le dépassement de capacité (Ovf) afin d'assurer une fiabilité dans la reconnaissance.

**La fonction d'activation :**

Pour la fonction d'activation « Tangente Hyperbolique Sigmoidale », nous avons choisi l'utilisation d'un générateur de fonctions appelé LUT (Look Up Table) qui se trouve dans les blocs logiques du FPGA. Cette LUT est une partie mémoire dont nous pouvons stocker les valeurs des échantillons uniformes prélevés de la fonction Tansig dans l'intervalle [-1, +1] afin d'obtenir une approximation plus précise.

**3.3.5.2. Le classifieur matériel :**

La réalisation matérielle du classifieur est composée de trois blocs similaires (Neurone1, Neurone2, Neurone3) de la couche cachée, un bloc Neurone4 de la couche de sortie et un comparateur.

Les valeurs des sept paramètres d'entrée arrivent en même temps aux neurones de la couche cachée. Dans cette dernière, le calcul des entrées pondérées s'effectue en parallèle, puis ils résultent trois valeurs en sortie qui doivent passer en même temps au neurone de la couche de sortie. Ensuite, la sortie de cette dernière traverse le bloc (comparateur) qui a pour rôle d'arrondir la valeur calculée vers 1 ou 0 afin de distinguer la classe. Nous avons nommé ce principe de calcul par **Parallèle-Parallèle**.

L'architecture matérielle CLNH-PP est purement combinatoire, donc lorsque tous les descripteurs se présentent à l'entrée du classifieur, nous aurons la classe en temps réel. Pour cela, nous avons ajouté des registres pour stocker les sept paramètres d'entrée, ces registres sont munis d'une horloge commune afin de synchroniser le classifieur neuronal.

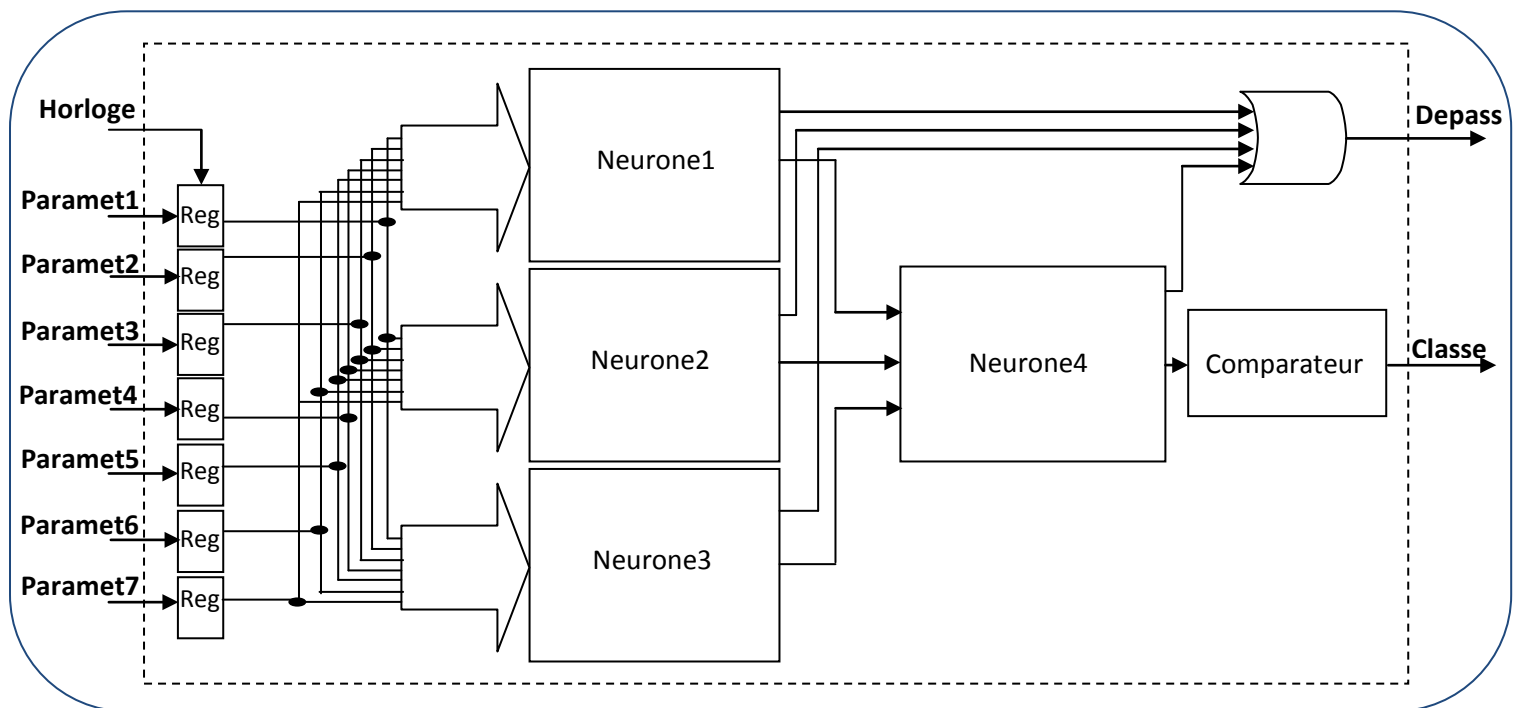


Figure 3.21. Architecture matérielle 'CLNH-PP' du classifieur neuronal.

### **3.4. Conclusion :**

Dans ce chapitre, nous avons présenté le classifieur neuronal sous l'environnement Matlab en montrant tous les points nécessaires pour la reconnaissance des arythmies cardiaques, en particulier les extrasystoles ventriculaires, puis nous avons entamé la réalisation matérielle de ce classifieur en utilisant deux principes de calcul : le Semi-Parallèle (SP) et le Parallèle-Parallèle (PP).

En utilisant les deux principes de calcul SP et PP, nous avons pu modifier plusieurs architectures matérielles déjà réalisées pour satisfaire les contraintes d'encombrement et du temps du cahier des charges afin d'avoir un classifieur neuronal en temps réel.

Maintenant que notre modèle neuronal est prêt et les architectures matérielles du classifieur CLN sont établies, nous présenterons dans le prochain chapitre les tests de nos prototypes et nous discuterons les différents résultats obtenus.



*Chapitre 4*

*L'implémentation du  
Classifieur Neuronal.*

## 4.1. Introduction :

Dans le chapitre 3, nous avons présenté le classifieur neuronal CLN pour la reconnaissance des extrasystoles ventriculaires ESV ainsi que leurs architectures matérielles (CLNH-SP, CLNH-PP). Ces dernières ont été réalisées en utilisant deux principes de calcul différents (SP, PP) afin de répondre aux contraintes d'encombrements et de temps dans le circuit reconfigurable.

Dans ce chapitre, nous allons suivre l'algorithme de conception proposé dans la figure 4.1 afin d'implémenter et évaluer l'efficacité des classifieurs CLNH-SP et CLNH-PP. nous présentons les trois implémentations réalisées en comparant leurs performances avec l'approche Software. Nous terminons ce chapitre par une étude comparative entre les architectures matérielles réalisées en termes de ressources et de temps de calcul.

## 4.2. Le flot de conception proposé :

Dans le cadre de ce travail, nous suivons les phases proposées dans la figure 4.1 afin d'évaluer l'efficacité de l'approche matérielle pour la reconnaissance des extrasystoles ventriculaires. Pour ce faire, nous présentons les résultats de simulation de trois implémentations matérielles étudiées auparavant (chapitre 3) du classifieur CLN.

- ✚ 1<sup>ère</sup> implémentation : Le classifieur CLNH-SP1 sans détection de dépassement de capacité.
- ✚ 2<sup>ème</sup> implémentation : Le classifieur CLNH-SP2 avec détection de dépassement de capacité.
- ✚ 3<sup>ème</sup> implémentation : Le classifieur CLNH-PP avec détection de dépassement de capacité.

Le cadre de notre implémentation matérielle nous a imposé d'établir quelques protocoles de calcul pour le prétraitement des données, des opérations.... Nous avons défini deux protocoles de calcul expérimentaux :

### ✚ Protocole1 :

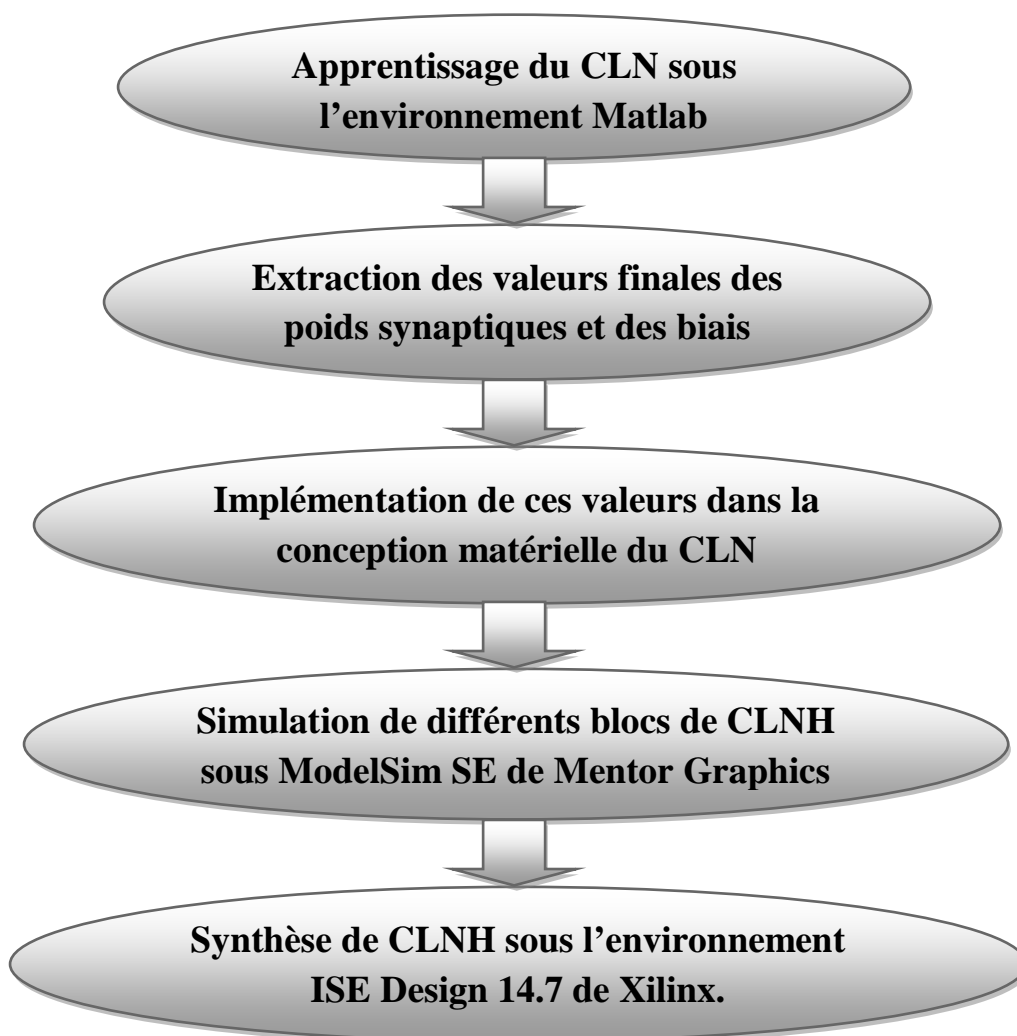
- La division des valeurs des poids synaptiques et des biais par un coefficient entraine des transformations qui peuvent fausser les résultats.
- Pour éviter les erreurs, nous multiplions la sortie de chaque Neurone\_NC par un coefficient, et puisque la sortie de l'étage est liée directement à la fonction d'activation,
- Alors, la multiplication par ce coefficient sera intégrée dans le calcul de la fonction d'activation sous forme de table de vérité (*True Table*).

Ce protocole de calcul a été utilisé d'une part dans la normalisation des valeurs des poids synaptiques et des biais, et d'autre part, pour éviter le dépassement de capacité dans les calculs.

**✚ Protocole2 :**

- La division des valeurs des paramètres d'entrée de la base d'apprentissage par un coefficient.
- Réaliser la phase d'apprentissage du classifieur sous l'environnement Matlab.
- Recueillir les valeurs finales des poids synaptiques et des biais, puis les normaliser en utilisant le Protocole1.
- La division des valeurs des paramètres d'entrée de la base de test par le même coefficient.
- Procéder au test des classifieurs réalisés sous les deux environnements.

Ce protocole de calcul a été utilisé pour la minimisation des valeurs des paramètres d'entrée du classifieur afin d'éviter le débordement dans les calculs.



**Figure 4.1. Schéma général du flot de conception.**

Selon l'algorithme de conception proposé pour chaque implémentation, nous commençons par le développement du classifieur CLN sous l'environnement Matlab en obtenant les valeurs finales des poids synaptiques et des biais qui seront utilisées dans les conceptions



matérielles réalisées. Ensuite, chaque bloc de ces architectures écrit en langage VHDL sera vérifié et simulé sous l'environnement ModelSimSe 6.2c (Figure 4.2).

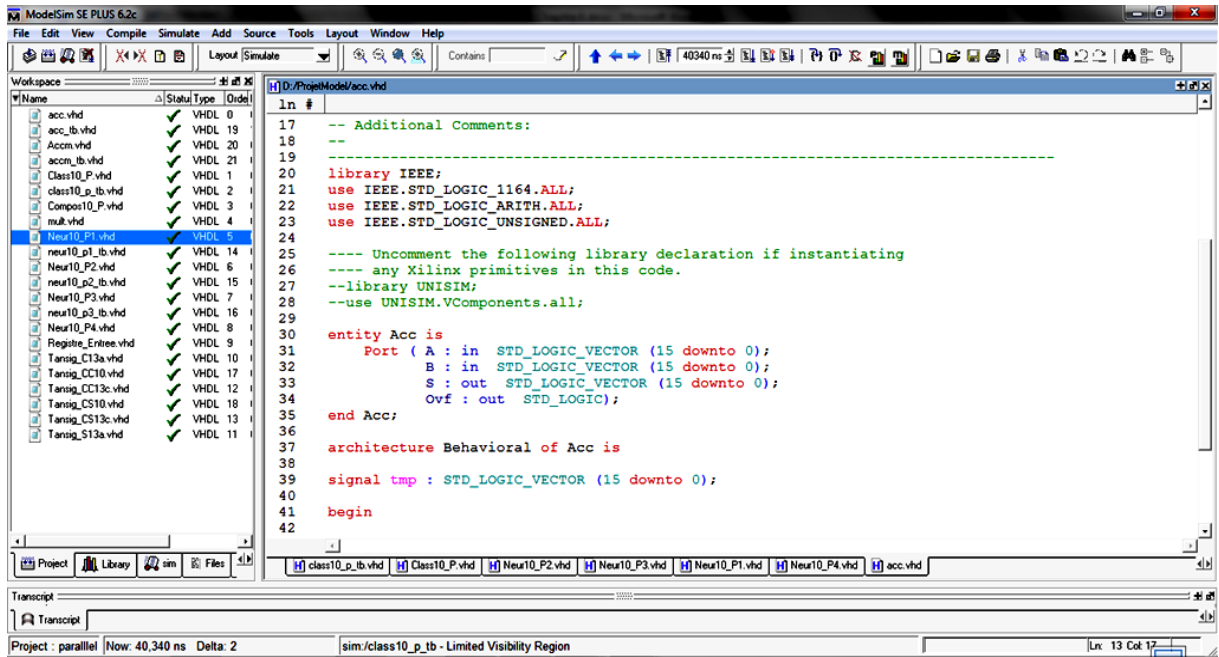
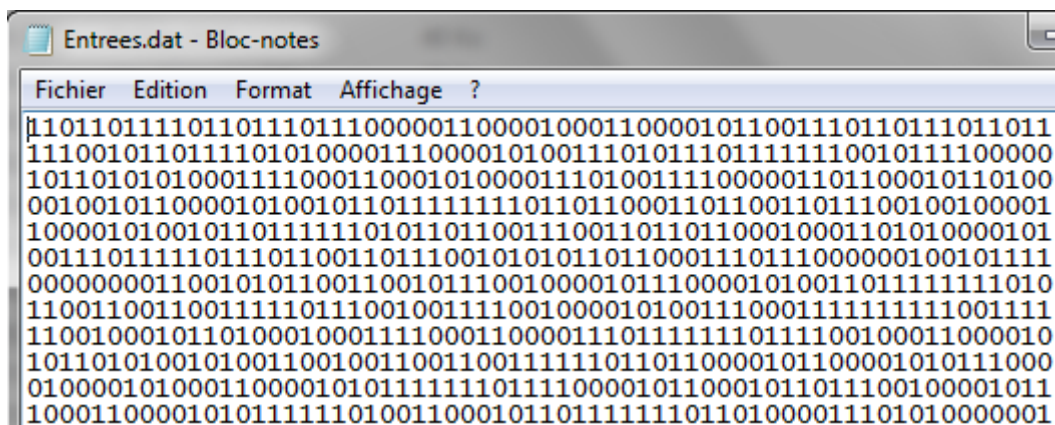


Figure 4.2. L'environnement ModelSimSe 6.2c.

La validation du fonctionnement de différentes architectures sous l'environnement ModelSim s'effectue en générant les paramètres d'entrée de chaque enregistrement de la base de test (proposée en chapitre 3) avec Matlab sous forme d'un fichier texte. Ce dernier contient les valeurs de données en binaires codées sur 1 octet chacune. En ajoutant des bibliothèques spécifiques, le langage VHDL peut lire les stimuli présentés dans les fichiers textes (les paramètres d'entrée du classifieur) et même écrire les résultats (la classe, les dépassements) dans d'autres fichiers textes (Figure 4.3).



(a)

Sorties.dat - Bloc-notes		Overflows.dat - Bloc-notes		
Fichier	Edition	Format	Affichage	?
CLASSE	TEMPS DE SIMULATION	Depas1	Depas2	TEMPS DE DETECTION
0	à t = 220 ns	0	0	0 ns
0	à t = 440 ns	0	0	20 ns
0	à t = 660 ns	0	0	40 ns
0	à t = 880 ns	0	0	60 ns
0	à t = 1100 ns	0	0	80 ns
0	à t = 1320 ns	0	0	100 ns
0	à t = 1540 ns	0	0	120 ns
0	à t = 1760 ns	0	0	140 ns
0	à t = 1980 ns	0	0	160 ns
0	à t = 2200 ns	0	0	180 ns
0	à t = 2420 ns	0	0	200 ns
0	à t = 2640 ns	0	0	220 ns
0	à t = 2860 ns	0	0	240 ns
0	à t = 3080 ns	0	0	260 ns
0	à t = 3300 ns	0	0	280 ns

(b)

**Figure 4.3. Les fichiers des stimules et des résultats du CLNH.**  
**(a). Fichier contenant les paramètres d'entrée. (b). Fichiers contenant les classes et les dépassements de capacité.**

Une fois les classifieurs développés et implémentés, leur performance sera évaluée en termes de taux de reconnaissance, de sensibilité et de spécificité d'une manière à comparer les performances entre les deux approches d'implémentation (*software* et *hardware*) du même classifieur.

Enfin, les classifieurs réalisés CLNH-SP2 et CLNH-PP seront synthétisés sous l'environnement ISE Design 14.7 afin de comparer les performances en termes de temps d'exécution et de ressources logiques utilisées dans le circuit reconfigurable FPGA. L'environnement ISE Design 14.7 de la société Xilinx (Figure 4.4) comporte les outils de CAO (Conception Assistée par Ordinateur) comme les outils de synthèse et simulation, les outils de placement et de routage.

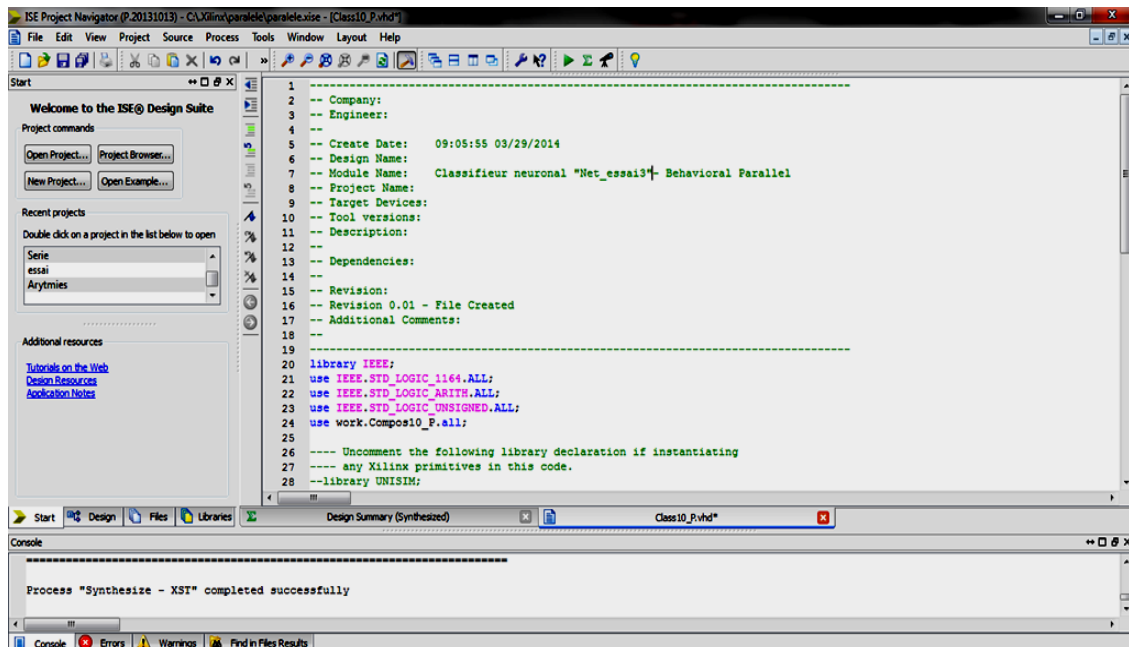


Figure 4.4. L'environnement ISE Design 14.7.

### 4.3. La première implémentation du classifieur neuronal CLN :

Dans cette section, nous présenterons les résultats de la première architecture matérielle réalisée du CLN qui utilise le principe de calcul (SP). La conception matérielle du classifieur CLNH-SP1 contient des additionneurs qui ne détectent pas les dépassements de capacité.

#### 4.3.1. Phase d'apprentissage du CLN :

L'apprentissage du classifieur CLN a été réalisé par l'algorithme de rétro-propagation du gradient. Nous avons mis les valeurs finales des poids synaptiques ainsi que du biais de chaque Neurone\_NC dans le tableau suivant :

Tableau 4.1. Les valeurs finales des poids synaptiques et des biais du CLN.

La couche cachée								
	W <sub>21</sub>	W <sub>22</sub>	W <sub>23</sub>	W <sub>24</sub>	W <sub>25</sub>	W <sub>26</sub>	W <sub>27</sub>	Biais
Neurone_NC1	0.7548	-2.5792	0.4918	0.2300	1.1569	-1.6393	0.9678	-1.0594
Neurone_NC2	-3.3731	0.5674	-0.4501	-1.8664	5.1861	-0.0805	1.0824	2.1944
Neurone_NC3	-0.5111	-0.5567	0.9087	-0.6610	0.7128	0.5871	-0.5284	-1.7004
La couche de sortie								
	W <sub>31</sub>	W <sub>32</sub>	W <sub>33</sub>	Biais				
Neurone_NC4	-3.2160	3.7280	1.5435	-1.8955				

#### 4.3.1.1. Les poids synaptiques et les biais du CLN:

Dans notre architecture matérielle, la représentation des données est codée sur 8 bits dans un espace qui varie entre -1 et +1. Le calcul effectué au niveau de chaque Neurone\_NC i est :

$$b_i + W_{i1} * P_1 + W_{i2} * P_2 + \dots + W_{ij} * P_j = a_i \quad (4.1)$$

Avec  $b_i$  et  $W_{ij}$  : les valeurs normalisées du biais et des poids synaptiques respectives pour un Neurone\_NC.

$P_j$  : les valeurs normalisées des paramètres d'entrée.

D'après le tableau 4.1, nous remarquons que les valeurs de quelques poids et biais sont en dehors de l'intervalle  $[-1, +1]$ , et afin de les normaliser, nous avons appliqué le premier protocole de calcul expérimental (**Protocole1**).

Nous divisons toutes les valeurs par un coefficient  $\lambda$  de l'équation 4.2 :

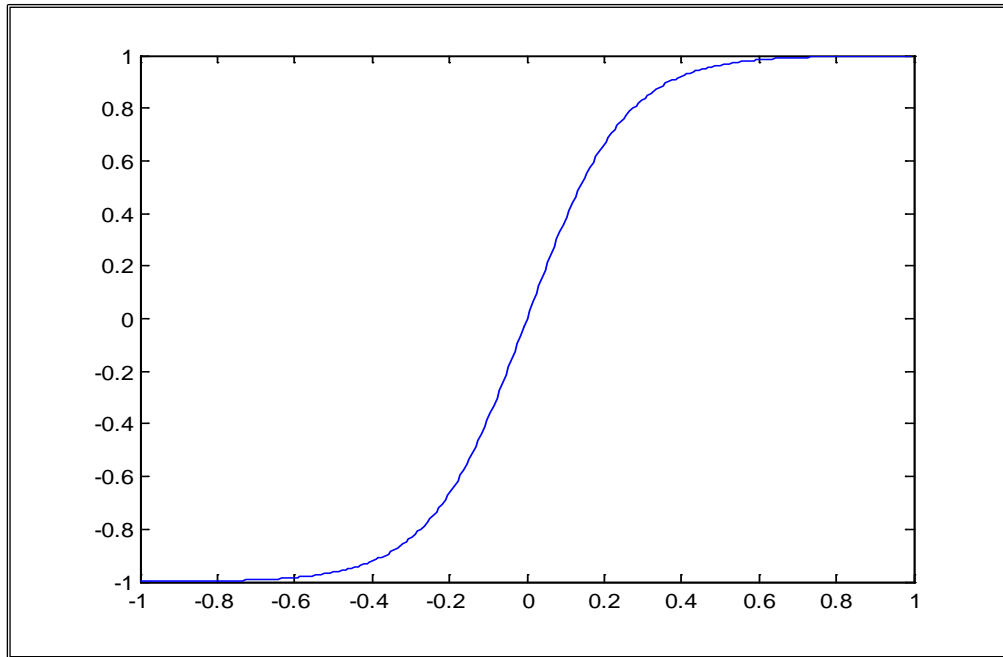
$$b'_i + W'_{i1} * P_1 + W'_{i2} * P_2 + \dots + W'_{ij} * P_j = a'_i \quad (4.2)$$

$$\text{Avec : } \begin{cases} b'_i = b_i / \lambda \\ W'_{ij} = W_{ij} / \lambda \\ a'_i = a_i / \lambda \end{cases} \quad (4.3)$$

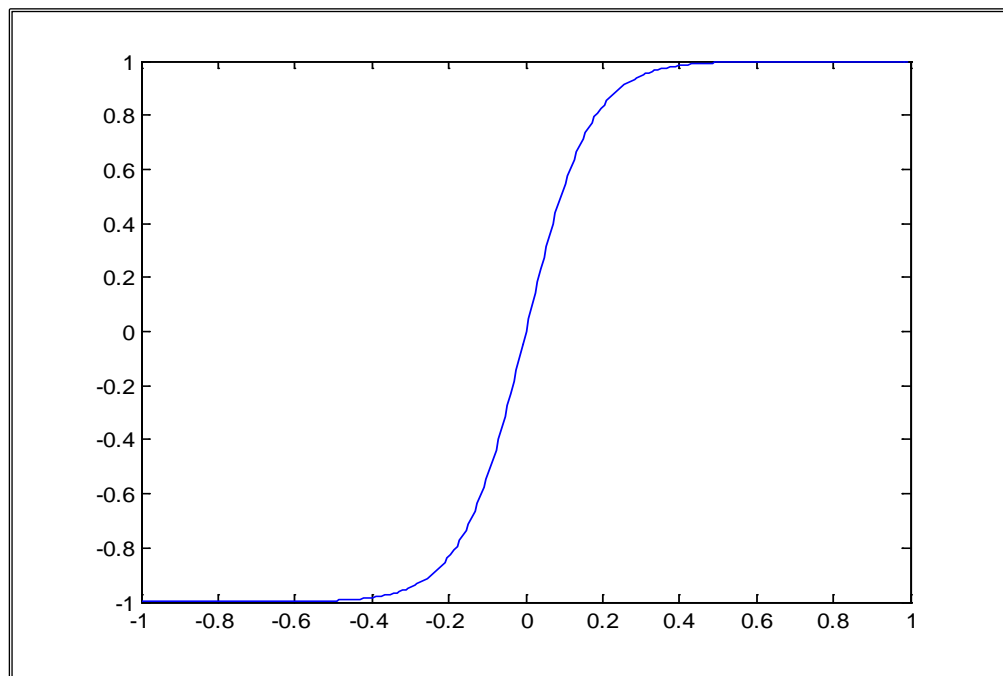
La division de ces valeurs par le coefficient  $\lambda$  entraîne des transformations qui risquent de fausser les résultats à la sortie de chaque Neurone\_NC. Alors, pour assurer de bons résultats, nous multiplions la sortie de chaque Neurone\_NC par ce coefficient  $\lambda$ , et comme la sortie de l'étage est liée directement à la fonction d'activation, la multiplication par  $\lambda$  sera intégrée dans le calcul de la fonction d'activation sous forme de table de vérité (*True Table*).

#### 4.3.1.2. Les fonctions d'activation :

Le choix des valeurs des coefficients  $\lambda_1, \lambda_2$  des deux couches du réseau (la couche cachée et la couche de sortie) sera fixé par rapport à la valeur la plus éloignée des limites de l'intervalle  $[-1, +1]$  qui sont  $W_{25} = 5.1861$  et  $W_{32} = 3.7280$  (Tableau 4.1). Donc, les coefficients sont  $\lambda_1 = 6$  et  $\lambda_2 = 4$ . Par conséquent, nous calculerons les fonctions d'activation  $\text{tansig}(6x)$  et  $\text{tansig}(4x)$  des deux couches respectivement (Figure 4.5).



(a)



(b)

**Figure 4.5. Les fonctions d'activation des deux couches du CLN.**  
(a). Tansig (6x) de la couche cachée. (b). Tansig (4x) de la couche de sortie.

#### 4.3.2. Le classifieur CLNH-SP1 :

Après l'implémentation de toutes les valeurs des poids synaptiques et des biais normalisés dans la conception matérielle du CLN [BABA\_10/2014], la simulation de quelques blocs du

classifieur CLNH-SP1 sous l'environnement ModelSim est présentée dans les figures suivantes :

- **La mémoire ROM :**

Nous avons stocké les valeurs des poids synaptiques de chaque neurone dans une mémoire du type ROM.

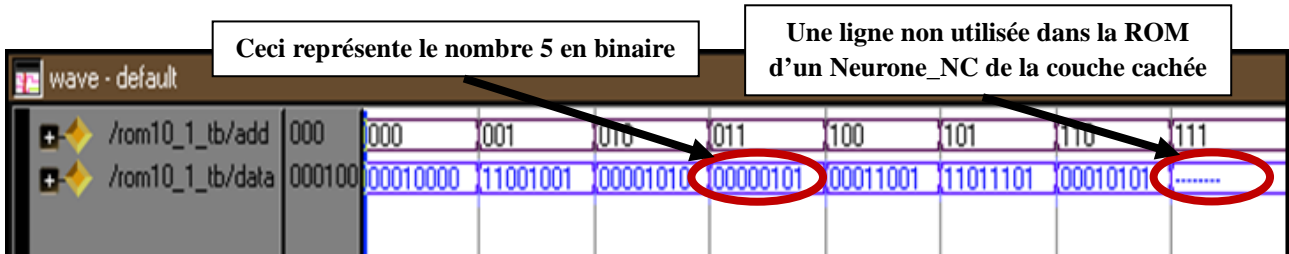


Figure 4.6. La simulation du bloc Rom sous l'environnement ModelSim.

- **Le multiplieur :**

Dans la figure 4.7, la multiplication de deux nombres binaires codés chacun sur 8 bits donne effectivement un résultat binaire codé sur 16 bits.

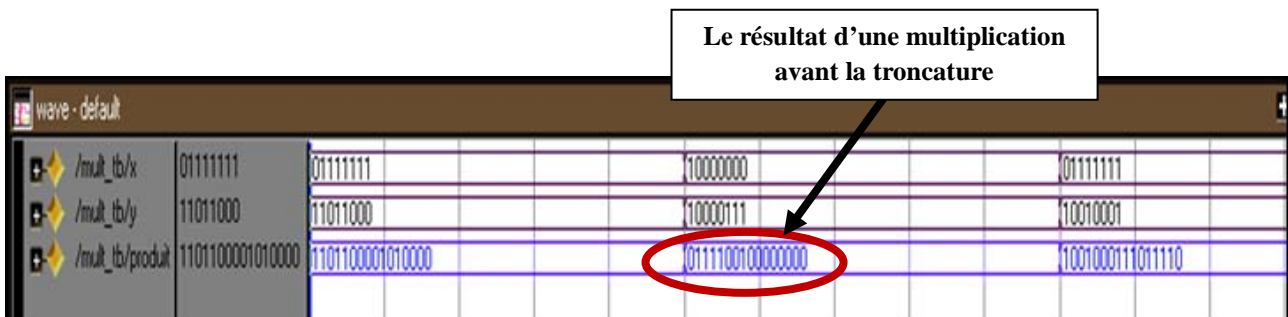


Figure 4.7. La simulation du bloc multiplieur sous l'environnement ModelSim.

- **L'additionneur :**

Après une simple troncature sur la sortie du multiplieur de façon à préserver le bit de signe, la donnée se présente sur 8 bits à l'entrée de l'additionneur. Mais, nous observons (Figure 4.8) des dépassements de capacité détectés au niveau de la première et de la deuxième opération d'addition.

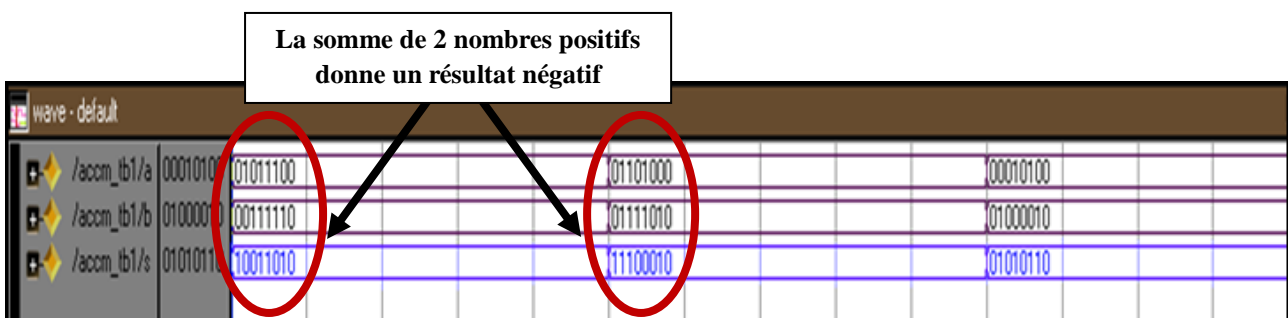
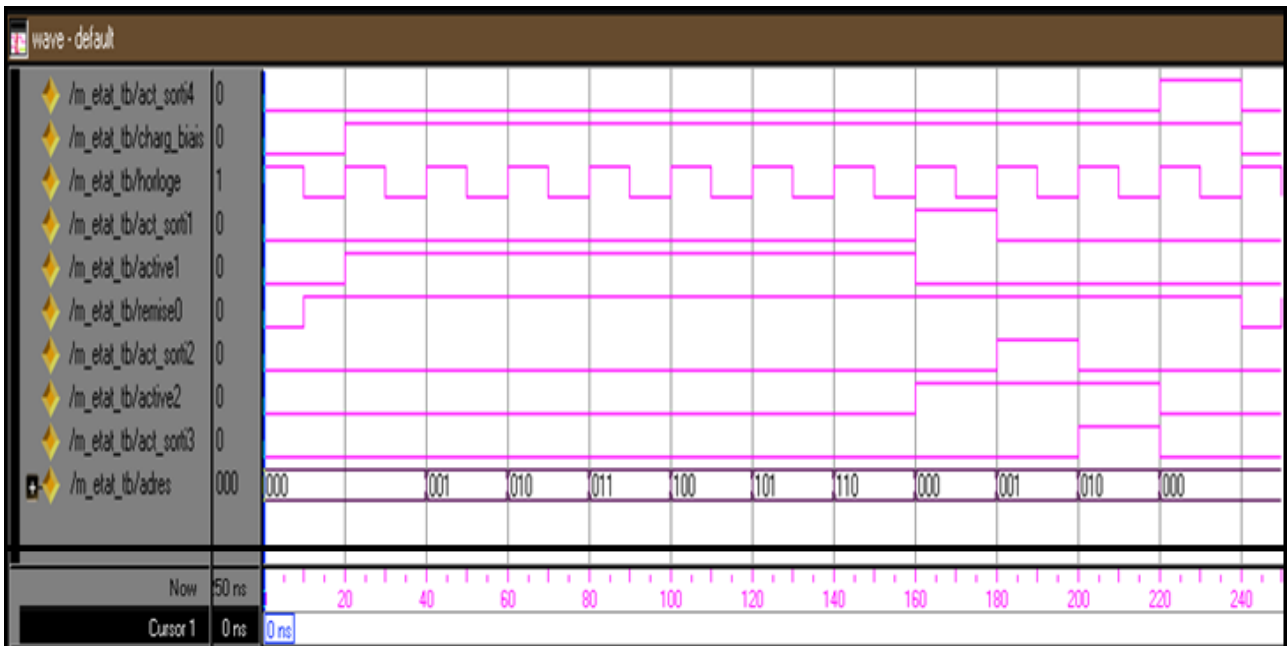


Figure 4.8. La simulation du bloc additionneur sous ModelSim.

- **Le séquenceur :**

La simulation du séquenceur (Figure 4.9) sous ModelSim est identique au chronogramme réalisé dans le chapitre précédent (chapitre 3).



**Figure 4.9. La simulation du bloc de séquenceur sous ModelSim.**

En se basant sur la représentation des données sur 8 bits et en utilisant l'additionneur décrit auparavant, nous constatons des dépassements de capacité qui peuvent erronés les calculs. Pour remédier à ce problème, nous appliquons une deuxième fois le **Protocole1** :

- ✚ Nous divisons les valeurs normalisées des poids synaptiques et du biais (Eq 4.2) sur le nombre d'accumulations  $\beta$  effectuées au niveau de chaque Neurone\_NC d'où l'équation :

$$b_i'' + W_{i1}'' * P_1 + W_{i2}'' * P_2 + \dots + W_{ij}'' * P_i = a_i'' \quad (4.4)$$

$$\text{Avec : } \begin{cases} b_i'' = b_i' / \beta \\ W_{ij}'' = W_{ij}' / \beta \\ a_i'' = a_i' / \beta \end{cases} \quad (4.5)$$

Dans la conception matérielle du classifieur CLNH-SP1, nous effectuons 8 accumulations au niveau de chaque Neurone\_NC de la couche cachée et 4 accumulations au niveau du Neurone\_NC de la couche de sortie, donc  $\beta_1 = 8$  et  $\beta_2 = 4$  des deux couches respectivement.

- ✚ La division des valeurs normalisées (les poids synaptiques et les biais) par les coefficients  $\beta_1$  et  $\beta_2$  (des deux couches respectivement).

- ✚ La multiplication par  $\beta_1$  et  $\beta_2$  sera intégrée dans le calcul des fonctions d'activation déjà étudiées (Tansig(6x), Tansig(4x)). Donc nous implémenterons les fonctions d'activation Tansig(48x) et Tansig(16x) (des deux couches respectivement) dans le classifieur CLNH-SP1.

### 4.3.3. Résultats et discussion :

La base de test utilisée pour les simulations des classifieurs CLN et CLNH-SP1 sous les environnements Matlab et ModelSim est celle décrite dans le chapitre 3. Dans cette base, nous avons utilisé six enregistrements de la base de données MIT-BIH.

Dans cette première partie de résultats, nous comparons les performances obtenues par les deux approches d'implémentation (*Software* et *hardware*) qui sont présentées dans le tableau 4.2.

**Tableau 4.2. Les performances des classifieurs CLN et CLNH-SP1.**

Sujets	106		116		119		208		213		221	
	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
NbR	1996	824	2351	1094	1976	840	2133	1265	2774	1131	1973	1073
TC (%)	98.96	40.85	98.74	45.95	99.95	42.49	83.22	49.36	97.54	39.77	81.70	44.43
TP	1496	521	2246	1040	1535	574	1151	754	2597	1021	1581	784
TN	500	303	105	54	441	266	982	511	177	110	392	289
FP	8	162	1	50	0	157	0	347	21	87	1	82
FN	1	851	3	1139	0	855	220	642	13	1123	237	1127

Avec : NbR: le nombre de cas reconnus correctement.

TC: le taux de reconnaissance correct.

TP: le nombre de battements réellement ESV (Extrasystole Ventriculaire) correctement reconnu (Vrai Positifs).

FP: le nombre de battements réellement non ESV reconnu comme ESV (Faux positifs).

TN: le nombre de battements réellement non ESV correctement reconnu (Vrai Négatifs).

FN: le nombre de battements réellement ESV reconnu comme non ESV (Faux Négatifs).

À partir du tableau 4.2, nous remarquons que toutes les performances obtenues par le classifieur CLNH-SP1 sont largement dégradées par rapport au classifieur CLN, nous constatons un taux de reconnaissance variant de 39.77% à 49.36% calculé par l'implémentation matérielle, comparant à un taux variant de 81.70% à 99.95% obtenu par l'implémentation software du classifieur neuronal. Ainsi, nous observons un nombre important de battements réellement ESV qui sont reconnus comme de battements normaux (nombre de FN très grand) par le classifieur CLNH-SP1, et cela peut générer un risque majeur sur la santé du patient.



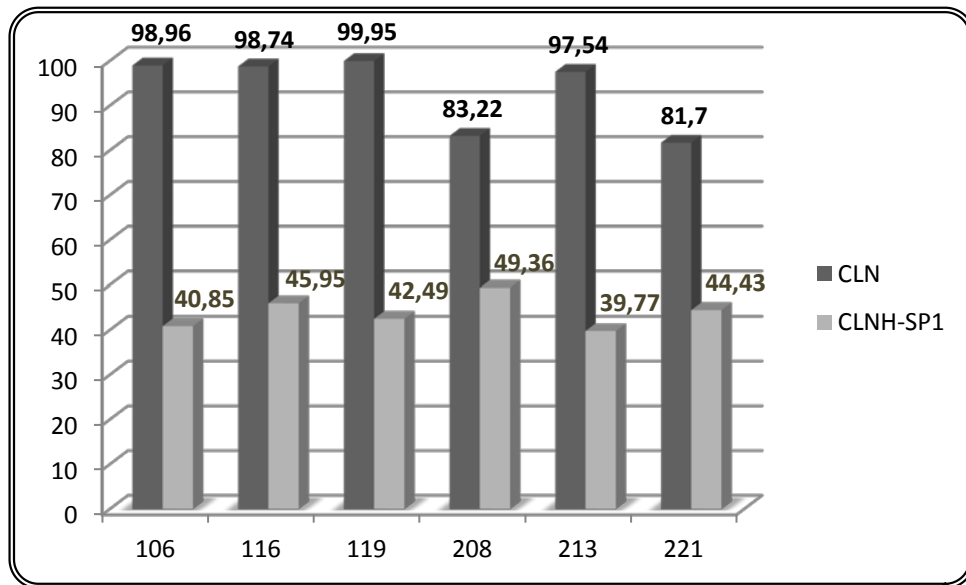


Figure 4.10. Le taux de classification obtenu des classifieurs CLN et CLNH-SP1.

Pour mieux évaluer notre classifieur, nous avons utilisé les grandeurs TP, TN, FP et FN afin de calculer certains paramètres statistiques comme la sensibilité (Se) et la spécificité (Sp), donnés par les équations suivantes :

$$Se = TP / (TP + FN) \quad (4.6)$$

$$Sp = TN / (TN + FP) \quad (4.7)$$

La **sensibilité** est la fraction de cas réel « ESV » correctement reconnu sur tous les cas réels ESV, tandis que la **spécificité** est la fraction de cas réel « N » (Normal) correctement rejeté sur tous les cas réels N.

Tableau 4.3. Le calcul de la sensibilité et de la spécificité des classifieurs CLN et CLNH-SP1.

Sujets	106		116		119		208		213		221	
	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
Se (%)	99.93	37.97	99.86	47.72	100	40.16	83.95	54.01	99.50	47.62	86.96	41.02
Sp (%)	98.42	65.16	99.05	51.92	100	62.88	100	59.55	89.39	55.83	99.74	77.89

Nous relevons du tableau 4.3 un taux de sensibilité et de spécificité très faible obtenu sous l'environnement ModelSim (Implémentation *Hardware* (Se : 37.97% à 54.01%) et (Sp : 51.92% à 77.89%)) par rapport au taux calculé sous l'environnement Matlab (Implantation *Software* (Se : 83.95% à 100%) et (Sp : 89.39% à 100%)) dans chaque enregistrement de la base de test. Donc le classifieur CLNH-SP1 a réalisé une mauvaise reconnaissance des cas malades ainsi que les cas normaux.

Après une étude approfondie des résultats, nous avons donc constaté que la solution établit auparavant (Protocole1) pour éviter le dépassement de capacité, entraine une diminution sensible du taux de reconnaissance, de la sensibilité et de la spécificité. C'est probablement à

cause de la perte importante de l'information qui a été provoqué par la division des entrées pondérées sur le nombre d'additions effectuées.

Afin d'améliorer les résultats de l'implémentation matérielle du classifieur CLN, nous avons procédé à la seconde implémentation qui détecte premièrement le nombre de dépassements de capacité provoqués, puis nous résoudrons ce problème avec une approche différente.

#### 4.4. La deuxième implémentation du classifieur CLN :

L'objectif principal de cette implémentation est de détecter le dépassement de capacité lors de la classification matérielle des données médicales [BABA\_2016]. La conception matérielle du classifieur CLNH-SP2 utilise le principe de calcul (SP) et elle contient des additionneurs détectant les dépassements de capacité.

##### 4.4.1. Le classifieur CLNH-SP2 :

Dans cette section, nous présentons la simulation du nouvel additionneur qui détecte le débordement de calcul afin de vérifier l'exactitude de nos résultats. Ainsi que nous expliquons d'autres blocs du classifieur CLNH-SP2.

- **Le nouveau bloc d'additionneurs :**

Nous observons dans la deuxième opération d'addition (Figure 4.11) que la somme de deux nombres positifs donne un nombre négatif codé chacun sur 8 bits, ce qui provoque un dépassement de capacité après 25 ns (Ovf = 1).

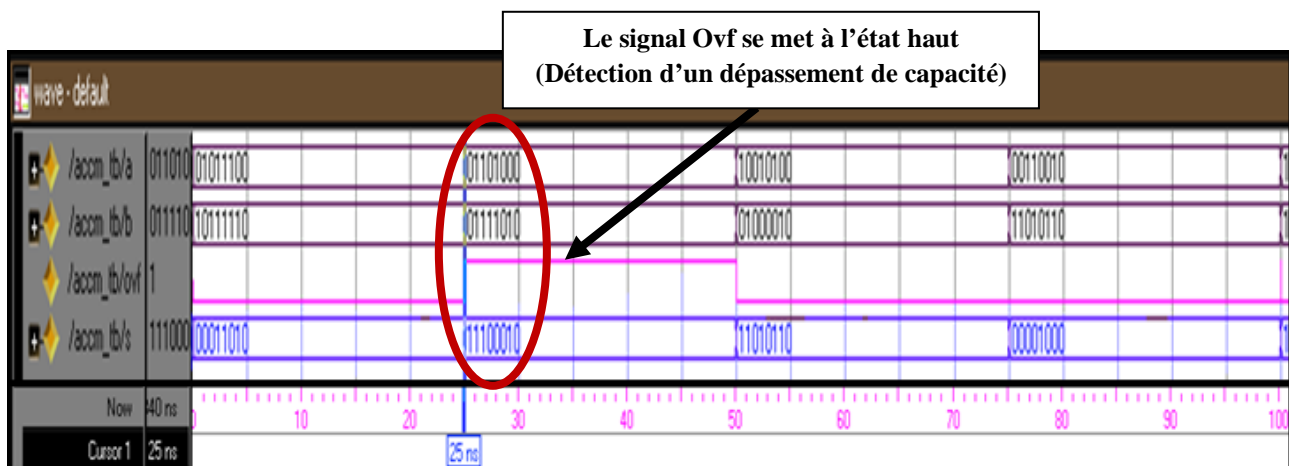


Figure 4.11. La simulation du nouveau additionneur sous ModelSim.

- **Le Neurone\_NC :**

La simulation du bloc Neurone\_NC sous l'environnement ModelSim est donnée par la figure 4.12, nous remarquons que la sortie du bloc est un vecteur codé sur 8 bits.

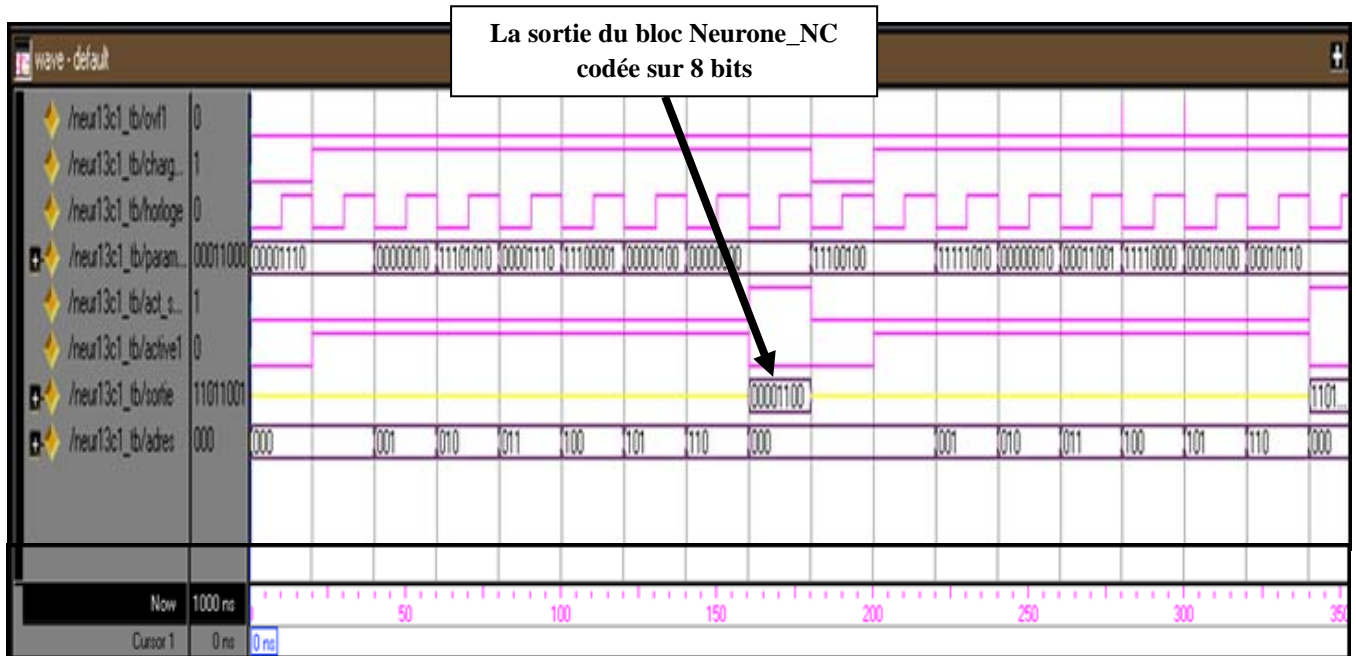


Figure 4.12. La simulation d'un Neurone\_NC de la couche cachée sous ModelSim.

- La couche cachée :

Nous observons dans la simulation de la couche cachée (Figure 4.13) sous l'environnement ModelSim les sorties des trois Neurone\_NC qui se mettent en série (après 160 ns), et cela montre le principe de calcul SP qui se base sur le parallélisme dans la même couche et la transmission des données séquentiellement vers la couche de sortie.

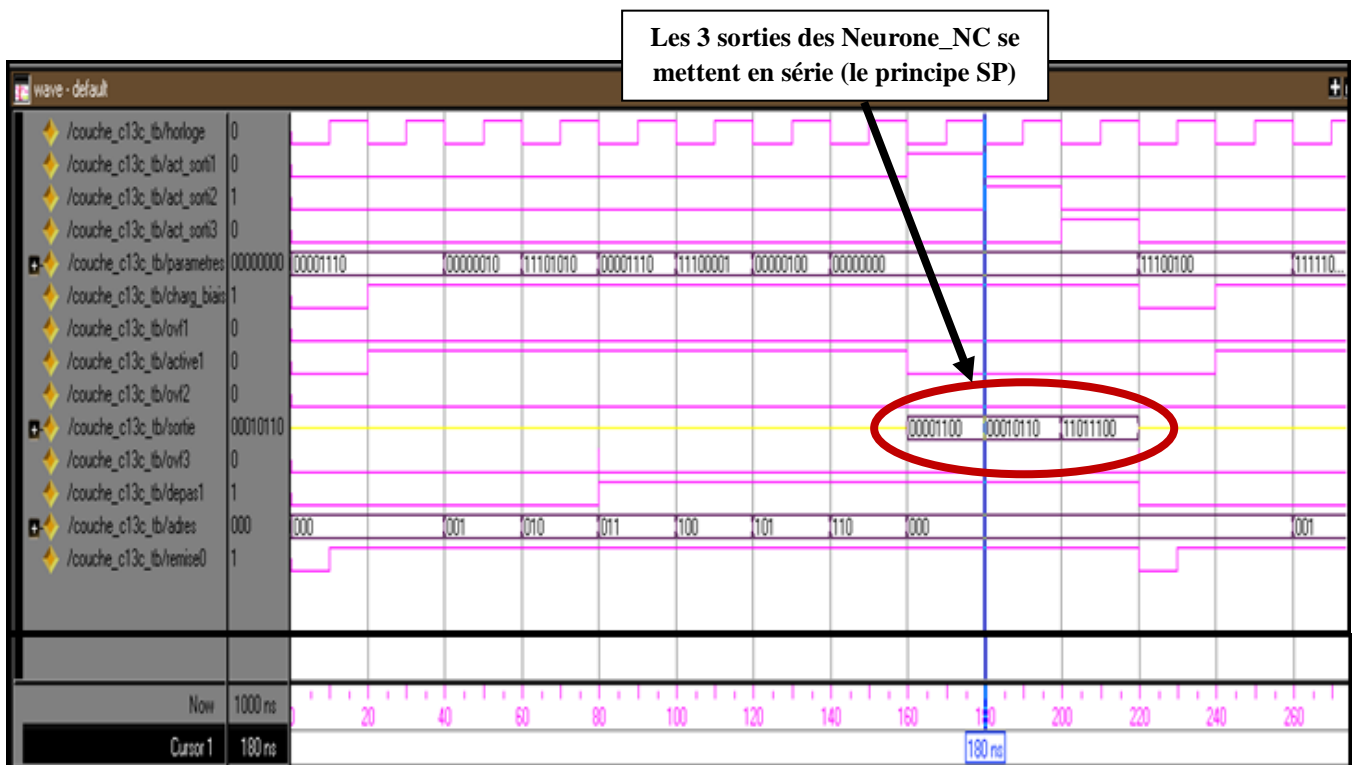


Figure 4.13. La simulation du bloc de la couche cachée sous ModelSim.

• **Le comparateur :**

En testant le bit MSB du vecteur de sortie dans le bloc comparateur, la simulation de ce dernier (Figure 4.14) présente les résultats suivants :

- Le signal à la sortie du comparateur se met à l'état haut si MSB = 0 (une détection d'une anomalie).
- Le signal à la sortie du comparateur se met à l'état bas si MSB = 1 (reconnaissance d'un cas normal).
- Le signal à la sortie du comparateur se met à l'état haut impédance (Z) si la détection se fait le moment de calcul.

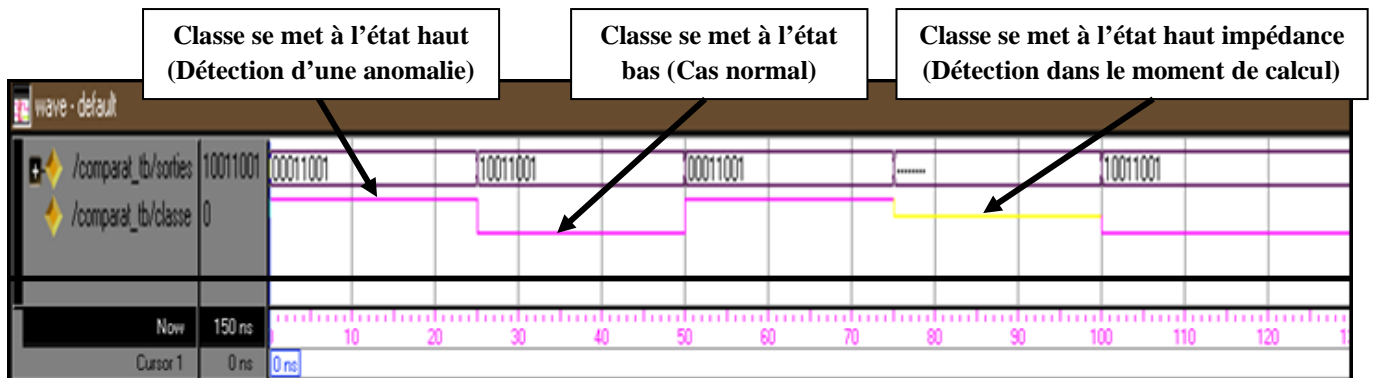


Figure 4.14. La simulation du bloc comparateur sous ModelSim.

**4.4.2. Résultats et discussion :**

Les performances des classifieurs CLN et CLNH-SP2 obtenues sont présentées dans le tableau suivant :

**Tableau 4.4. Les performances des classifieurs CLN et CLNH-SP2.**

Record	106		116		119		208		213		221	
	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
NbR	1996	1996	2351	2350	1976	1816	2133	2130	2774	2771	1973	1970
TC (%)	98.96	98.96	98.74	98.70	99.95	91.86	83.22	83.11	97.54	97.43	81.70	81.57
TP	1496	1497	2246	2245	1535	1505	1151	1155	2597	2598	1581	1585
TN	500	499	105	105	441	311	982	975	177	173	392	385
FP	8	9	1	1	0	129	0	9	21	26	1	8
FN	1	1	3	3	0	29	220	217	13	12	237	227
OvF	/	110	/	8	/	777	/	95	/	15	/	37

Avec : OvF : le nombre de dépassements de capacité détectés (*Overflows*).

D'après le tableau 4.4, le taux de reconnaissance, TP, TN, FP et FN sont très similaires pour les deux approches d'implémentation sauf au niveau du sujet 119 où les résultats sont différents (29 battements réellement ESV reconnu comme non ESV et 129 battements réellement non ESV reconnu comme ESV). Nous observons dans l'implémentation Hardware une détection importante de dépassement de capacité variant de 8 à 777 dans les différents

enregistrements de la base de test. Alors, le calcul effectué par le classifieur CLNH-SP2 est erroné malgré la similarité du taux de reconnaissance avec le classifieur CLN (Figure 4.15).

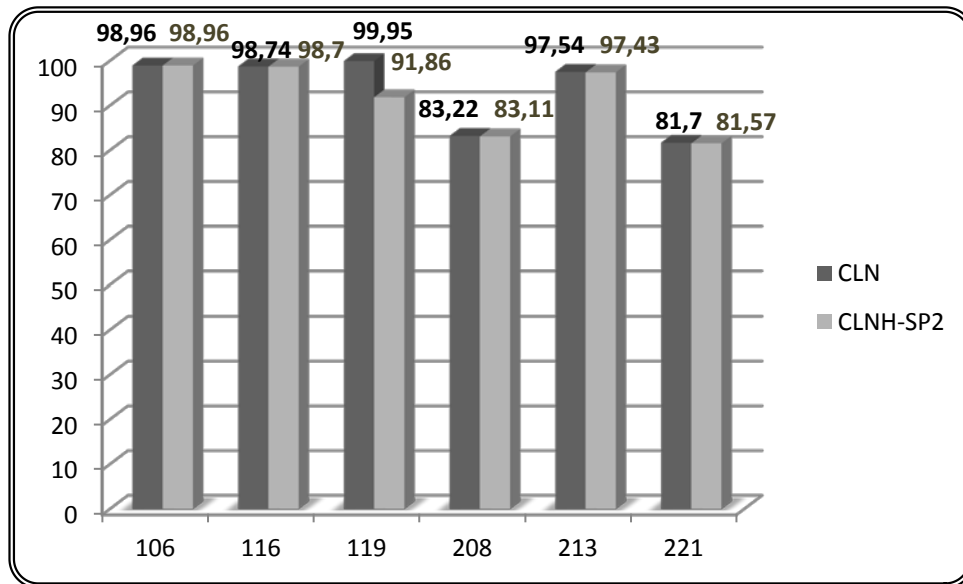


Figure 4.15. Le taux de classification obtenu des classifieurs CLN et CLNH-SP2.

Dans cette seconde implémentation, nous avons présenté les résultats de simulation de l'approche matérielle en utilisant le sujet 119.

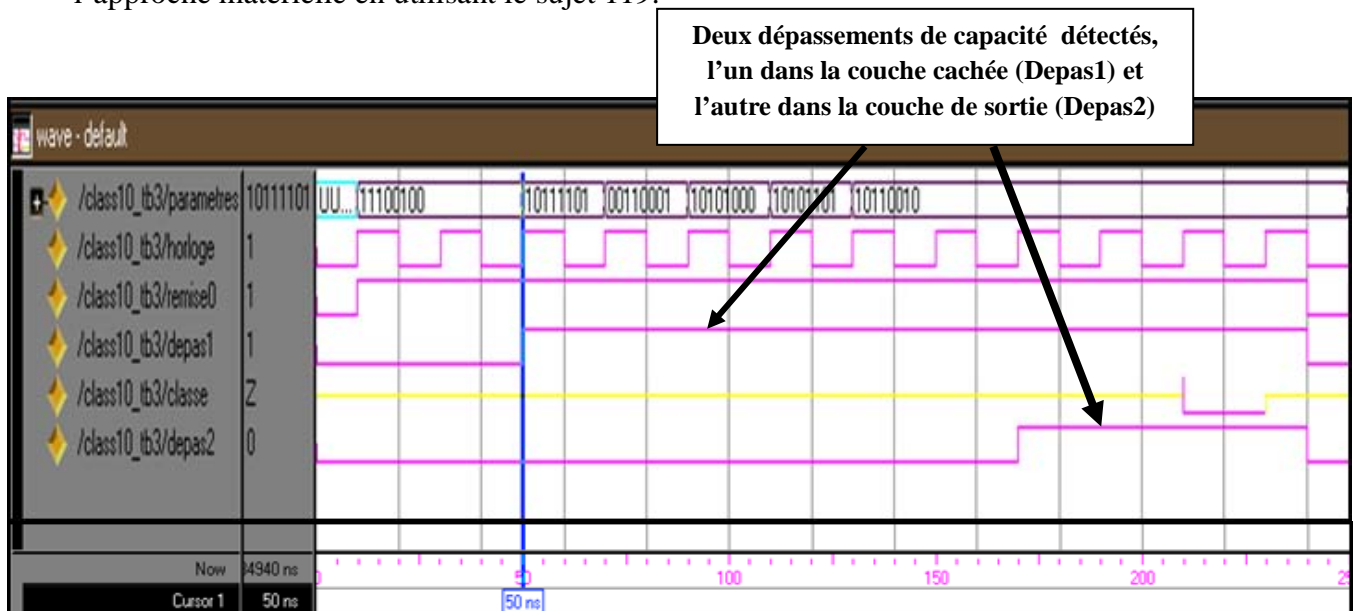


Figure 4.16. La simulation du classifieur CLNH-SP2 sous ModelSim.

Lors de la simulation du classifieur CLNH-SP2 sous ModelSim, nous avons détecté deux dépassements de capacité dans la classification du premier cycle cardiaque du sujet 119 (Figure 4.16), le premier (Depas1) a été remarqué dans la couche cachée à l'instant 50 ns et le second (Depas2) dans la couche de sortie à l'instant 170 ns. La détection de ces débordements a été maintenue jusqu'à la fin de la classification du 1<sup>er</sup> cycle cardiaque afin de les noter dans le fichier texte.

Donc l'utilisation du classifieur CLNH-SP2 a provoqué un débordement important dans le calcul. Pour remédier à ce problème, nous appliquons un autre protocole de calcul expérimental nommé **Protocole2**. Ce dernier minimise les valeurs des paramètres d'entrée du classifieur afin que le résultat de la multiplication des poids synaptiques avec ces paramètres ne coïncide pas avec les limites de l'intervalle  $[-1, +1]$ , et l'accumulation de ces entrées pondérées avec le biais ne provoque pas un dépassement de capacité.

La minimisation des valeurs des paramètres d'entrée est une opération de division par un coefficient  $K$  dont le choix de sa valeur est empirique. Cependant, l'emploi du concept Protocole2 exige une autre phase d'apprentissage du classifieur sous l'environnement Matlab nommé CLN1 et un autre classifieur matériel CLNH1-SP2, puisque pour ces derniers, les valeurs des paramètres d'entrée sont divisées par le coefficient  $K$ .

#### 4.4.3. Le classifieur CLNH1-SP2 :

L'objectif principal du classifieur CLNH1-SP2 est de réduire au maximum le nombre de dépassements de capacité détectés par le 1<sup>er</sup> classifieur CLNH-SP2. Pour ce faire, nous avons choisi d'augmenter la valeur du coefficient  $K$  par un pas  $\alpha$  ( $\alpha = 0.2$ ), et cela à partir de  $1+\alpha$ . L'intérêt majeur de cette démarche est de gagner en précision. Cependant, après plusieurs tests, nous avons choisi  $K=2$  afin de faciliter la conception matérielle de la division en remplaçant cette dernière par un décalage à droite.

##### 4.4.3.1. Phase d'apprentissage du classifieur CLN1 :

Selon le **Protocole2**, nous avons divisé les paramètres d'entrée de la base d'apprentissage par le coefficient  $K=2$  et nous avons relancé l'apprentissage du classifieur CLN1 sous l'environnement Matlab. Nous avons noté les valeurs finales des poids synaptiques et des biais dans le tableau suivant :

**Tableau 4.5. Les valeurs finales des poids synaptiques et des biais du CLN1.**

La couche cachée								
	$W_{21}$	$W_{22}$	$W_{23}$	$W_{24}$	$W_{25}$	$W_{26}$	$W_{27}$	Biais
Neurone_NC1	-1.8722	-1.7362	-1.3356	-1.0966	-0.3139	0.8990	0.4434	1.7680
Neurone_NC2	1.4393	-4.8258	0.3824	2.7269	0.3755	-3.8076	0.3002	-1.7648
Neurone_NC3	1.5170	3.4346	-2.1581	-0.8286	-6.2071	3.1316	-0.4886	-1.2160
La couche de sortie								
	$W_{31}$		$W_{32}$		$W_{33}$		Biais	
Neurone_NC4	-1.5663		-4.7127		-4.3522		-0.6318	

Dans l'implémentation matérielle, nous avons normalisé les valeurs des poids synaptiques et des biais en utilisant le concept **Protocole1**. Les valeurs  $W_{25} = -6.2071$  et  $W_{32} = -4.7127$  ont été choisis pour fixer la valeur des coefficients  $\lambda_1 = 7$  et  $\lambda_2 = 5$ . Ces dernières valeurs ont été utilisées pour calculer les fonctions d'activation de deux couches du classifieur respectivement (tansig (7x) de la couche cachée et tansig (5x) de la couche de sortie).

4.4.3.2. Résultats et discussion :

Pour évaluer les performances des deux classifieurs CLN1 et CLNH1-SP2 (Tableau 4.6), nous avons divisé les valeurs des paramètres d'entrée de la base de test par K=2.

Tableau 4.6. Les performances des classifieurs CLN1 et CLNH1-SP2.

Sujets	106		116		119		208		213		221	
K=2	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
NbR	1981	1983	2355	2350	1969	1966	2381	2354	2524	2457	2133	2093
TC (%)	98.21	98.31	98.91	98.70	99.60	99.44	92.90	91.85	88.75	86.40	88.32	86.67
TP	1496	1496	2250	2244	1527	1524	1396	1369	2326	2256	1743	1703
TN	485	487	105	106	442	442	985	985	198	201	390	390
FP	6	5	1	1	0	0	0	0	6	6	2	1
FN	0	0	6	7	0	0	59	67	39	67	51	57
Ovf	/	0	/	0	/	2	/	0	/	0	/	0

À partir des résultats présentés dans le tableau 4.6, nous constatons que les performances obtenues par les deux approches d'implémentation (Software et Hardware) sont presque similaires pour chaque sujet de la base de test.

Au niveau des sujets 208, 213 et 221, nous observons un nombre assez important de battements réellement ESV reconnus comme de battements normaux (nombre de FN élevé), variant de 57 à 67 par le classifieur CLNH1-SP2 malgré la disparition totale de dépassement de capacité. Cependant, nous remarquons au niveau du sujet 119 une bonne reconnaissance des extrasystoles ventriculaires (le classifieur a reconnu tous les cas ESV (FN=0)) avec un taux de classification de 99.44% bien qu'il y a une détection très faible de dépassement de capacité (Ovf=2). Pour le reste des sujets (106 et 116), un taux de classification correct, variant de 98.31% à 98.70% avec une élimination totale de débordement.

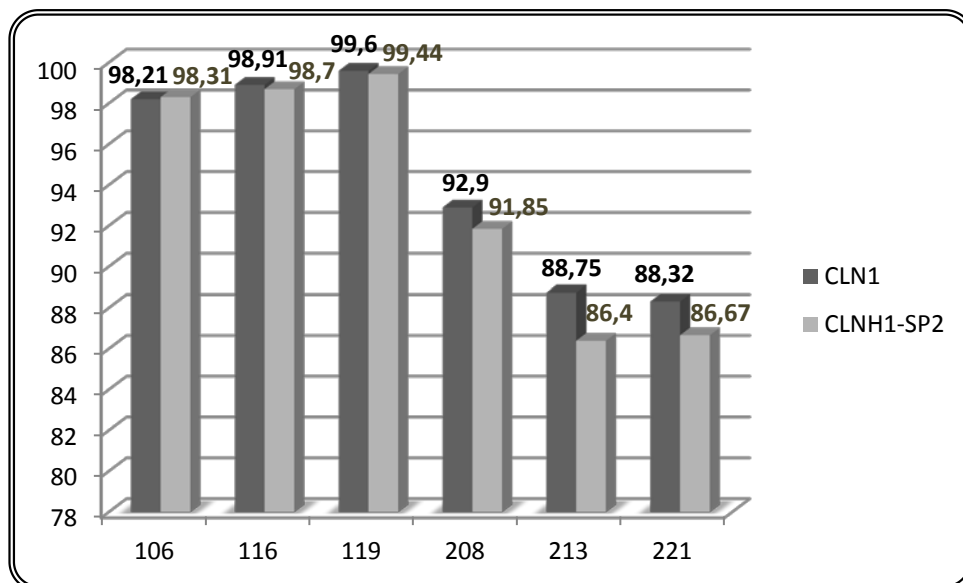
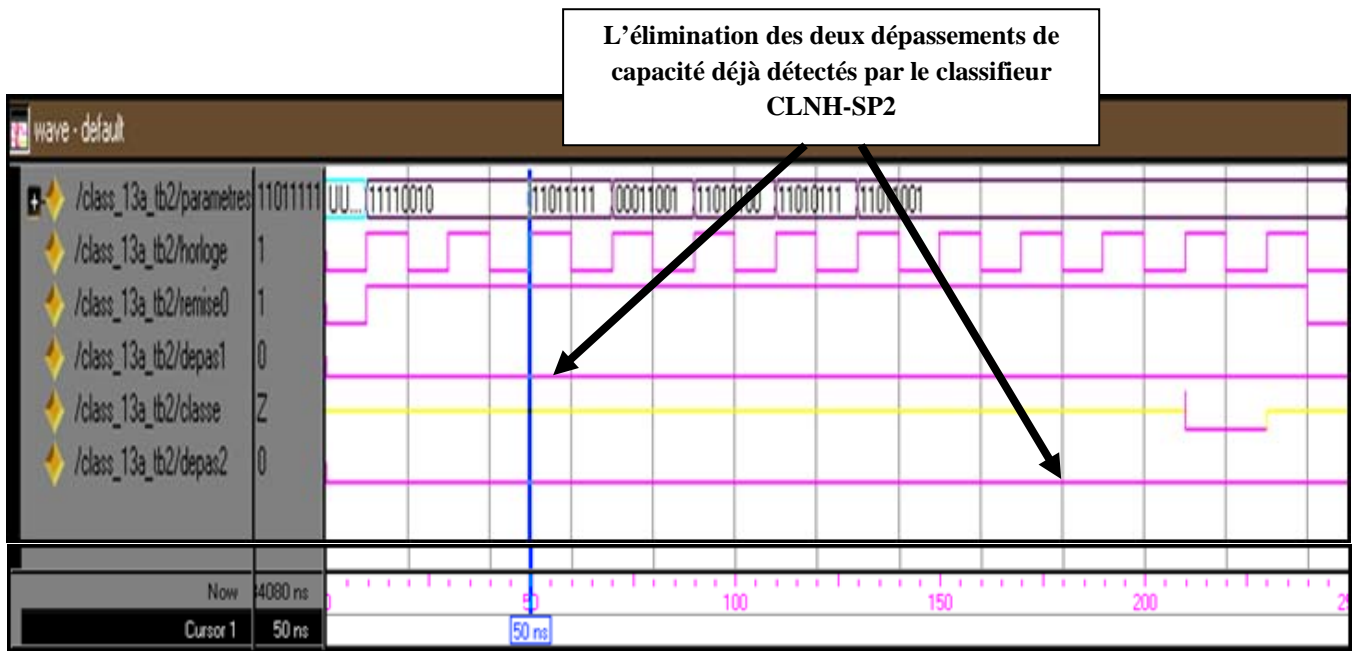
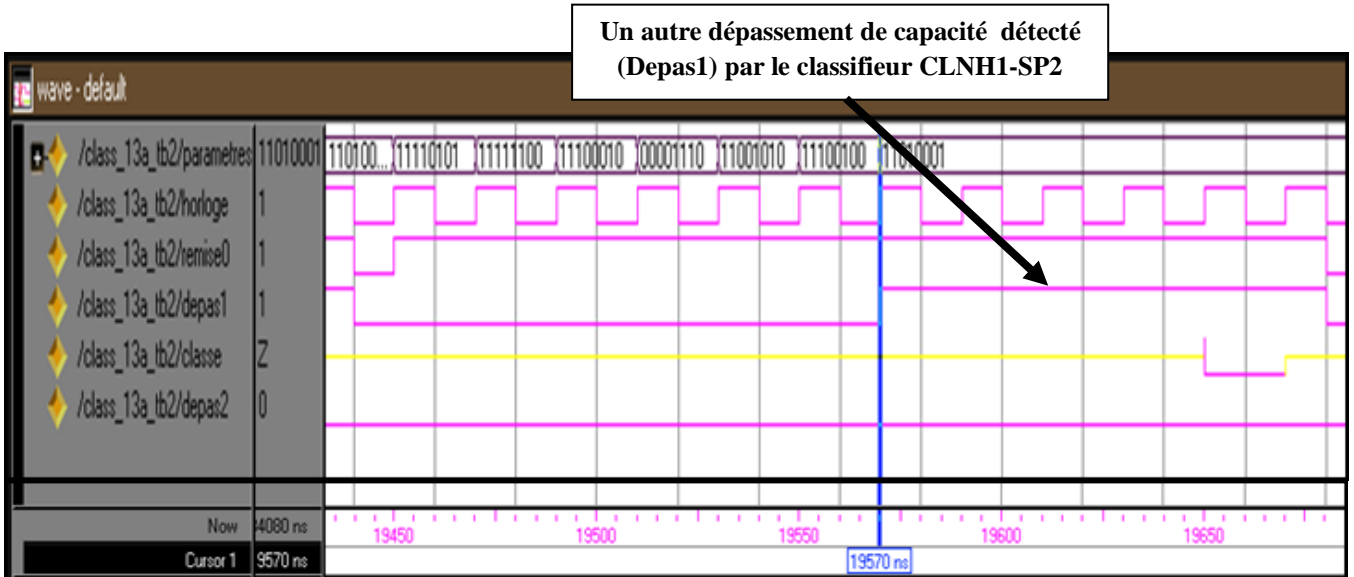


Figure 4.17. Le taux de classification obtenu des classifieurs CLN1 et CLNH1-SP2.

La simulation du classifieur CLNH1-SP2 sous ModelSim (Figure 4.18 (a)) montre que les deux dépassements provoqués lors de la classification du premier cycle cardiaque du sujet 119 à l'instant 50 ns et 170 ns ont été éliminés. Nous observons dans la figure 4.18 (b) une détection d'un autre débordement de calcul (Depas1) à l'instant 19750 ns.



(a)



(b)

**Figure 4.18. La simulation du classifieur CLNH1-SP2 sous ModelSim.**  
 (a). Le test du 1<sup>er</sup> cycle cardiaque du sujet 119. (b). Le test du 88<sup>ème</sup> cycle cardiaque du sujet 119.

Afin d'éliminer tous les dépassements de capacité provoqués par le classifieur CLNH1-SP2, nous avons réutilisé le concept du **Protocole2** en choisissant la valeur du coefficient  $K=4$  où nous avons nommé le classifieur par CLN2 et le classifieur matériel par CLNH2-SP2.



#### 4.4.4. Le classifieur CLNH2-SP :

L'objectif principal du classifieur CLNH2-SP2 est d'éliminer le dépassement de capacité détecté au niveau du sujet 119. Pour ce faire, nous avons choisi la valeur du coefficient  $K=4$ .

##### 4.4.4.1. Phase d'apprentissage du classifieur CLN2 :

Selon le **Protocole2**, nous avons divisé les paramètres d'entrée de la base d'apprentissage par le coefficient  $K=4$  et nous avons relancé l'apprentissage du classifieur CLN2 sous l'environnement Matlab. Nous avons noté les valeurs finales des poids synaptiques et des biais dans le tableau suivant :

**Tableau 4.7. Les valeurs finales des poids synaptiques et des biais du CLN2.**

La couche cachée								
	$W_{21}$	$W_{22}$	$W_{23}$	$W_{24}$	$W_{25}$	$W_{26}$	$W_{27}$	Biais
Neurone_NC1	4.4836	1.7954	-2.3343	-0.9930	4.4596	-7.5647	-6.8448	1.4460
Neurone_NC2	0.2572	6.6054	1.8368	-1.2393	-2.1945	3.7965	-7.1725	1.4143
Neurone_NC3	-6.5095	-0.9015	-2.8743	-2.2458	9.5166	-3.9973	2.6839	0.1598
La couche de sortie								
	$W_{31}$		$W_{32}$		$W_{33}$		Biais	
Neurone_NC4	-4.9307		5.6116		7.8327		-1.8329	

Nous avons normalisé les valeurs des poids synaptiques et des biais en choisissant les valeurs des coefficients  $\lambda_1 = 10$ ,  $\lambda_2 = 8$  des deux couches du classifieur (la couche cachée et la couche de sortie respectivement). Ce choix a été fixé par rapport aux valeurs  $W_{25} = 9.5166$  et  $W_{33} = 7.8327$  respectivement (Tableau 4.7), puis nous avons calculé les fonctions d'activation  $\text{tansig}(10x)$  et  $\text{tansig}(8x)$  de deux couches respectivement.

##### 4.4.4.2. Résultats et discussion :

Afin d'obtenir les performances des deux classifieurs CLN2 et CLNH2-SP2 (Tableau 4.8), nous avons divisé les valeurs des paramètres d'entrée de la base de test par  $K=4$ .

**Tableau 4.8. Les performances des classifieurs CLN2 et CLNH2-SP2.**

Sujets	106		116		119		208		213		221	
K=4	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
NbR	1915	1950	2031	2162	1977	1977	2003	2103	2779	2754	2098	2185
TC(%)	94.94	96.68	85.30	90.80	100	100	78.15	82.05	97.71	96.84	86.87	90.48
TP	1454	1497	1925	2057	1535	1535	1018	1118	2615	2617	1707	1794
TN	461	453	106	105	442	442	985	985	164	137	391	391
FP	29	30	1	1	0	0	0	0	32	49	1	2
FN	0	0	29	25	0	0	245	165	4	3	139	78
OvF	/	0	/	0	/	0	/	0	/	0	/	0

D'après le tableau 4.8, nous observons d'une part une dégradation dans le taux de classification obtenu par l'approche Software par rapport au taux calculé par l'approche Hardware au niveau des sujets 106, 116, 208 et 221. D'autre part, un taux de reconnaissance correct et similaire dans les deux approches d'implémentation (100%) au niveau du sujet 119.

Cependant, nous remarquons au niveau des sujets 116, 208, 213 et 221 que le nombre de battements réellement ESV reconnus comme de battements normaux est plus élevé sous Matlab à celui présenté sous ModelSim (FN<sub>208</sub>=245 par rapport à FN<sub>H208</sub>=165 (sous ModelSim), FN<sub>221</sub>=139 qui sont plus élevés de FN<sub>H221</sub>=78). Donc notre classifieur matériel CLNH2-SP2 (l'approche Hardware) a pu minimiser le nombre des cas malades non reconnus par rapport au classifieur CLN2 (l'approche Software).

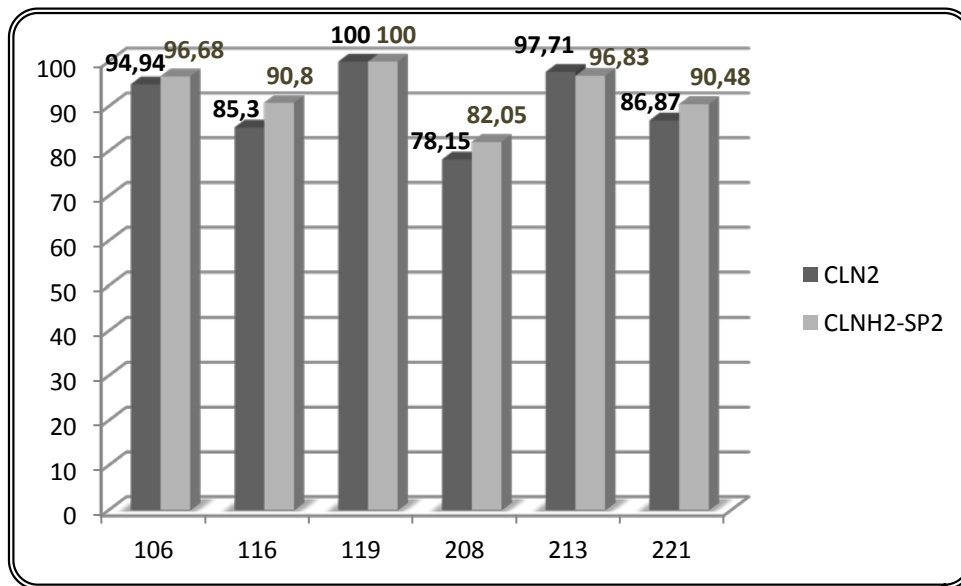


Figure 4.19. Le taux de classification obtenu des classifieurs CLN2 et CLNH2-SP2.

Le classifieur CLNH2-SP2 a réalisé un taux de classification meilleur par rapport au classifieur CLN2 (Figure 4.19) sur plusieurs enregistrements (106, 116, 208, 221).

Tableau 4.9. Le calcul de la sensibilité et de la spécificité des classifieurs CLN2 et CLNH2-SP2.

Record	106		116		119		208		213		221	
	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
Se (%)	100	100	98.51	98.79	100	100	80.60	87.14	99.84	99.88	92.47	95.83
Sp (%)	94.08	93.78	99.06	99.06	100	100	100	100	83.67	73.65	99.74	99.49

Nous constatons sur le tableau 4.9 un taux de sensibilité et de spécificité similaire obtenu par les deux approches d'implémentation, le taux de Se variant de 80.60% à 100% et le taux de Sp variant de 73.65% à 100%.

Nous observons dans la figure 4.20 que le classifieur CLNH2-SP2 a éliminé le dépassement de capacité détecté auparavant par le classifieur CLNH1-SP2 à l'instant 19570 ns.

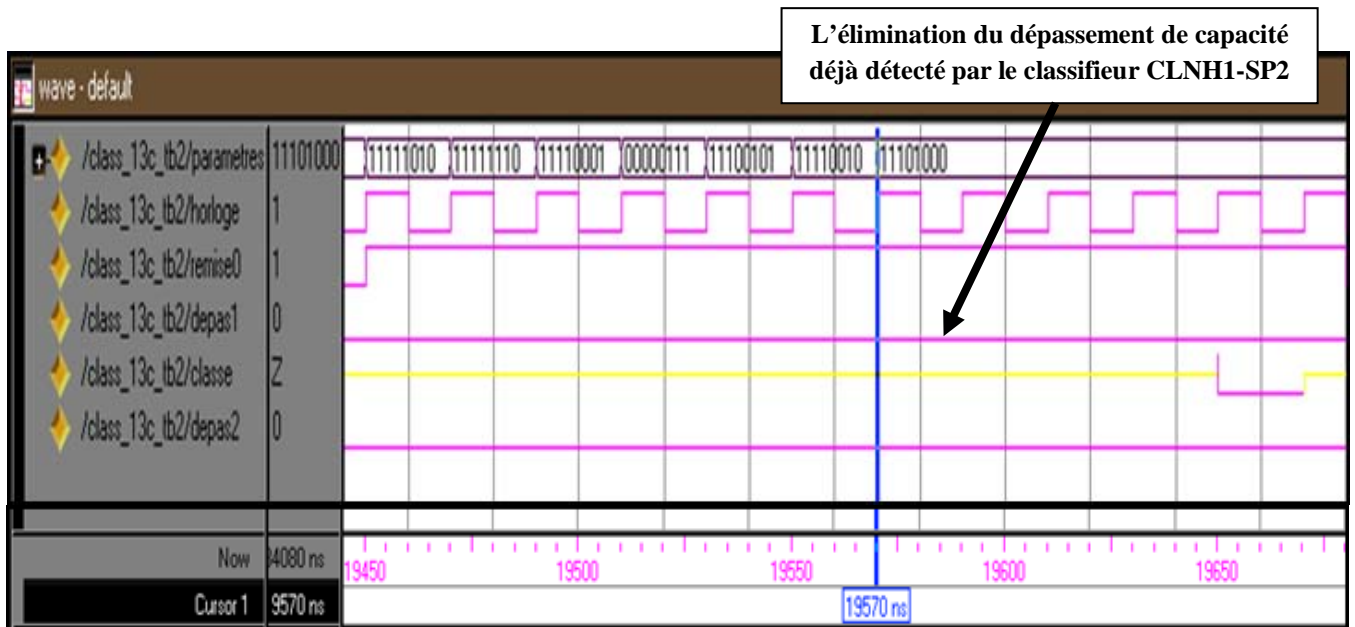


Figure 4.20. La simulation du classifieur CLNH2-SP2 du sujet 119 sous l'environnement ModelSim.

#### 4.5. La troisième implémentation du classifieur CLN :

Dans cette section, nous présenterons les résultats de la deuxième architecture matérielle réalisée du CLN qui utilise le principe de calcul (PP). La conception matérielle du classifieur CLNH-PP utilise quelques blocs réalisés dans la première architecture matérielle comme le multiplieur de Booth, l'additionneur avec détection de dépassement de capacité, la fonction d'activation et le comparateur.

##### 4.5.1. Le classifieur CLNH-PP :

Après l'implémentation de toutes les valeurs des poids synaptiques et des biais normalisées dans la nouvelle conception matérielle du CLN, nous avons simulé le bloc Neurone du classifieur CLNH-PP sous l'environnement ModelSim.

- **Le neurone :**

Le neurone est composé de blocs purement combinatoires comme les multiplieurs, les additionneurs, la fonction d'activation et un bloc qui contient les valeurs des poids synaptiques et du biais correspondant à ce neurone. Donc lorsque les données se présentent à l'entrée du bloc, nous obtenons la sortie à l'instant même (Figure 4.21).

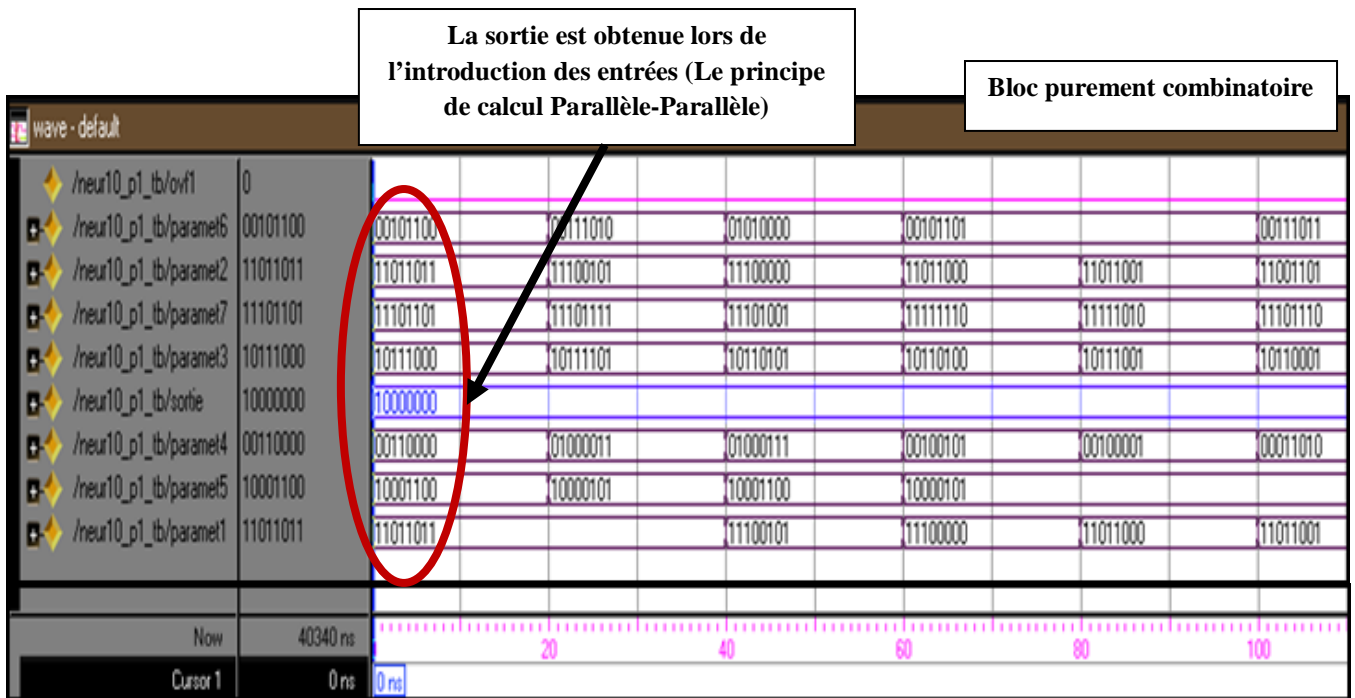


Figure 4.21. La simulation du bloc Neurone sous l'environnement ModelSim.

Afin de faire une étude comparative entre les deux architectures réalisées (paragraphe 4.7), nous avons établi la simulation de la nouvelle conception matérielle en choisissant les mêmes classifieurs utilisés dans la 2<sup>ème</sup> implémentation (CLN, CLN1 et CLN2).

• Résultats et discussion :

Les performances des classifieurs CLN et CLNH-PP sont présentées dans le tableau suivant :

Tableau 4.10. Les performances des classifieurs CLN et CLNH-PP.

Sujets	106		116		119		208		213		221	
	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
NbR	1996	1996	2351	2349	1976	1815	2133	2129	2774	2770	1973	1969
TC (%)	98.96	98.96	98.74	98.66	99.95	91.81	83.22	83.07	97.54	97.40	81.70	81.53
TP	1496	1497	2246	2245	1535	1505	1151	1155	2597	2598	1581	1585
TN	500	499	105	104	441	310	982	974	177	172	392	384
FP	8	10	1	2	0	130	0	10	21	267	1	9
FN	1	1	3	4	0	29	220	217	13	12	237	227
OvF	/	199	/	10	/	364	/	9	/	4	/	28

Au niveau de chaque sujet de la base de test (Tableau 4.10), nous remarquons une similarité du taux de classification obtenu des deux approches d'implémentation (Software et Hardware), malgré le nombre important de dépassements de capacité détecté par le classifieur CLNH-PP (un nombre qui varie de 4 à 364 débordements). En effet, le calcul effectué par l'approche matérielle est erroné.

Cependant, nous observons aussi au niveau des sujets 208 et 221 que le nombre de battements réellement ESV reconnus comme de battements normaux est très élevé sous les deux environnements, variant de 217 à 237 cas (FN très élevé).

Par l'approche Software, nous constatons un taux de classification (Figure 4.22) variant de 81.70% à 99.95% par rapport à un taux variant de 81.53% à 98.96% obtenu par l'approche Hardware.

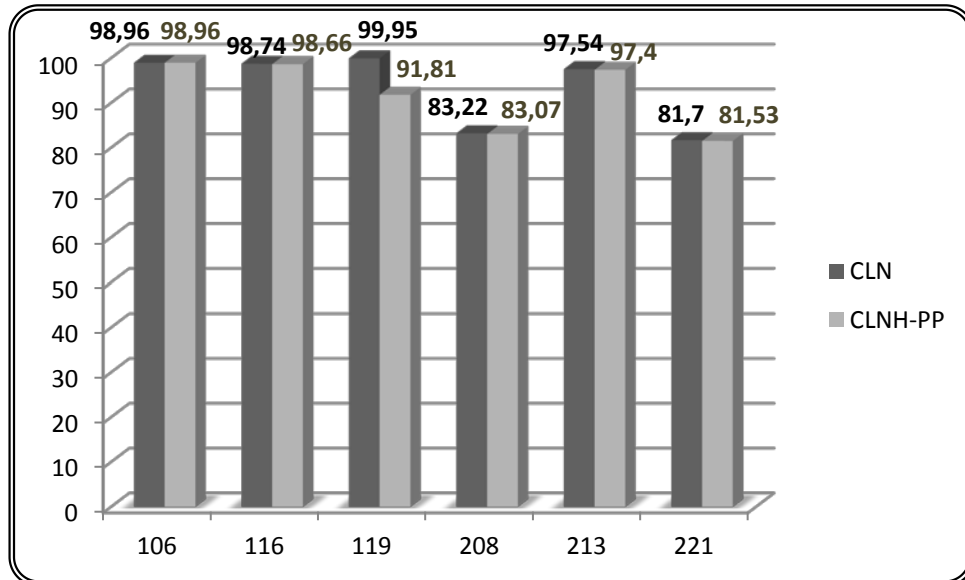


Figure 4.22. Le taux de classification obtenu des classifieurs CLN et CLNH-PP.

Dans cette troisième implémentation, nous avons choisi de présenter les simulations de l'approche matérielle en utilisant les cycles cardiaques du sujet 106.

Un dépassement de capacité détecté (Depass) par le classifieur CLNH-PP

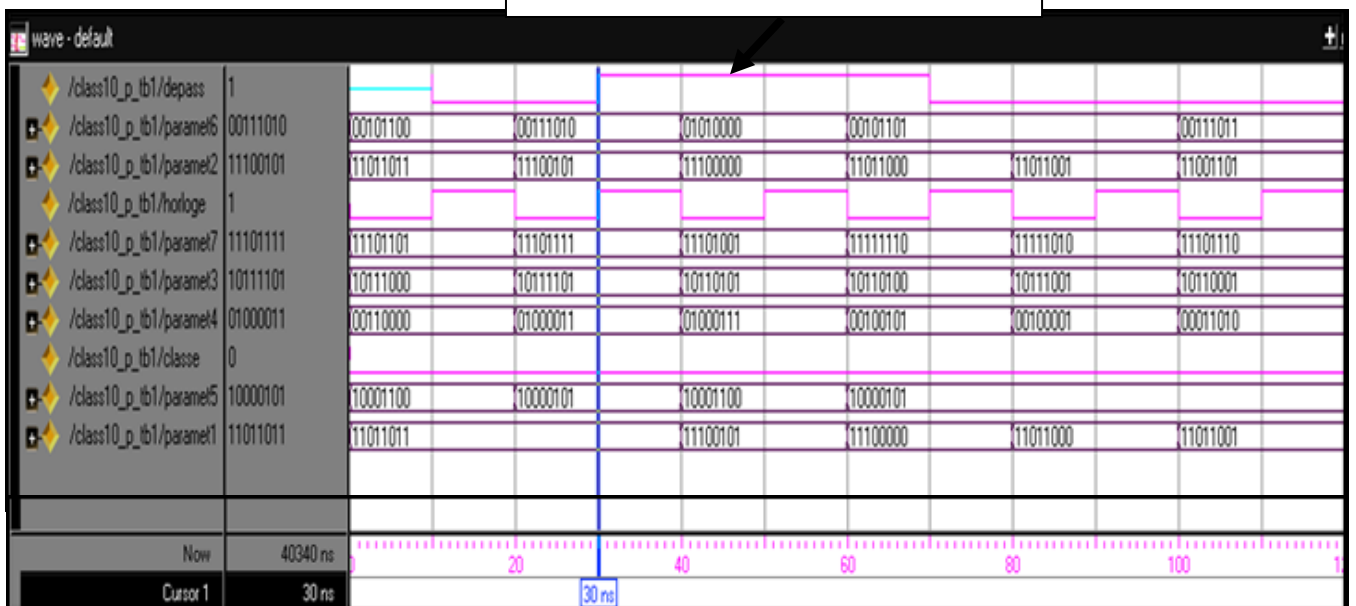


Figure 4.23. La simulation du classifieur CLNH-PP sous ModelSim.

La simulation du classifieur CLNH-PP est présentée sur la figure 4.23, dont nous observons un dépassement de capacité (Depass) détecté dans la classification du 2<sup>ème</sup> et du 3<sup>ème</sup> cycle cardiaque du sujet 106.

#### 4.5.2. Le classifieur CLNH1-PP :

L'objectif principal de cette partie est de réduire au maximum le nombre de dépassements de capacité détectés par le 1<sup>er</sup> classifieur CLNH-PP. Pour ce faire, nous avons implémenté matériellement le classifieur CLN1 réalisé dans la 2<sup>ème</sup> section.

- **Résultats et discussion :**

Pour évaluer les performances des deux classifieurs CLN1 et CLNH1-PP (Tableau 4.11), nous avons divisé les valeurs des paramètres d'entrée de la base de test par K=2.

**Tableau 4.11. Les performances des classifieurs CLN1 et CLNH1-PP.**

Record	106		116		119		208		213		221	
K=2	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
NbR	1981	1388	2355	2273	1969	1081	2381	1481	2524	2308	2133	1779
TC (%)	98.22	68.82	98.91	95.46	99.60	54.68	92.90	57.78	88.75	81.15	88.32	73.66
TP	1496	1256	2250	2192	1527	1055	1396	1473	2326	2220	1743	1777
TN	485	132	105	81	442	26	985	8	198	201	390	2
FP	6	381	1	27	0	416	0	977	6	130	2	391
FN	0	230	6	46	0	443	59	80	39	295	51	198
OvF	/	675	/	72	/	1006	/	1018	/	196	/	467

D'après le tableau 4.11, le classifieur CLNH1-PP entraîne un nombre très élevé de dépassement de capacité (jusqu'à 1018 débordements) par rapport au classifieur CLNH-PP pour les différents sujets. Donc, l'implémentation du classifieur CLN1 dans la seconde conception matérielle donne des mauvais résultats. Ainsi, nous remarquons un faible taux de reconnaissance obtenu par l'approche Hardware variant de 54.68% à 95.46% par rapport au taux calculé par l'approche Software variant de 88.32% à 99.60% (Figure 4.24).

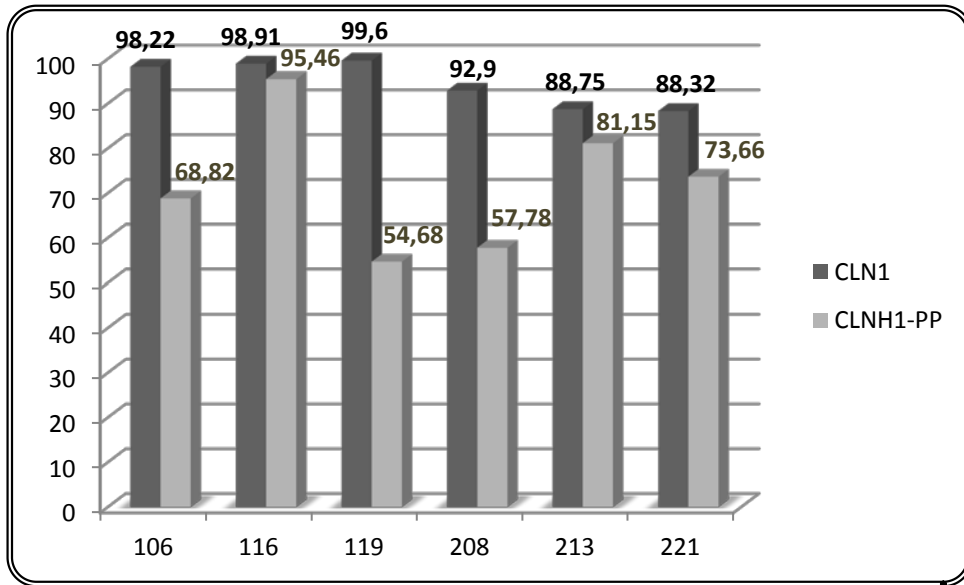
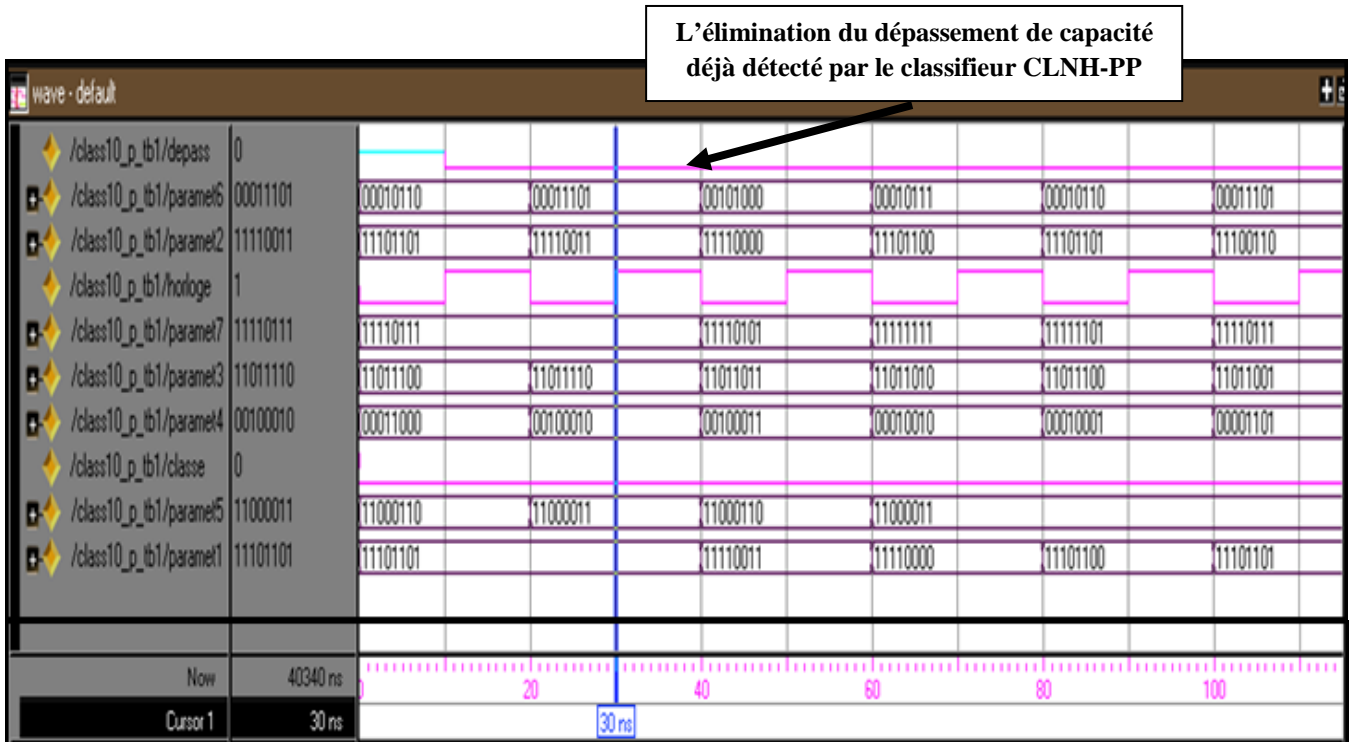
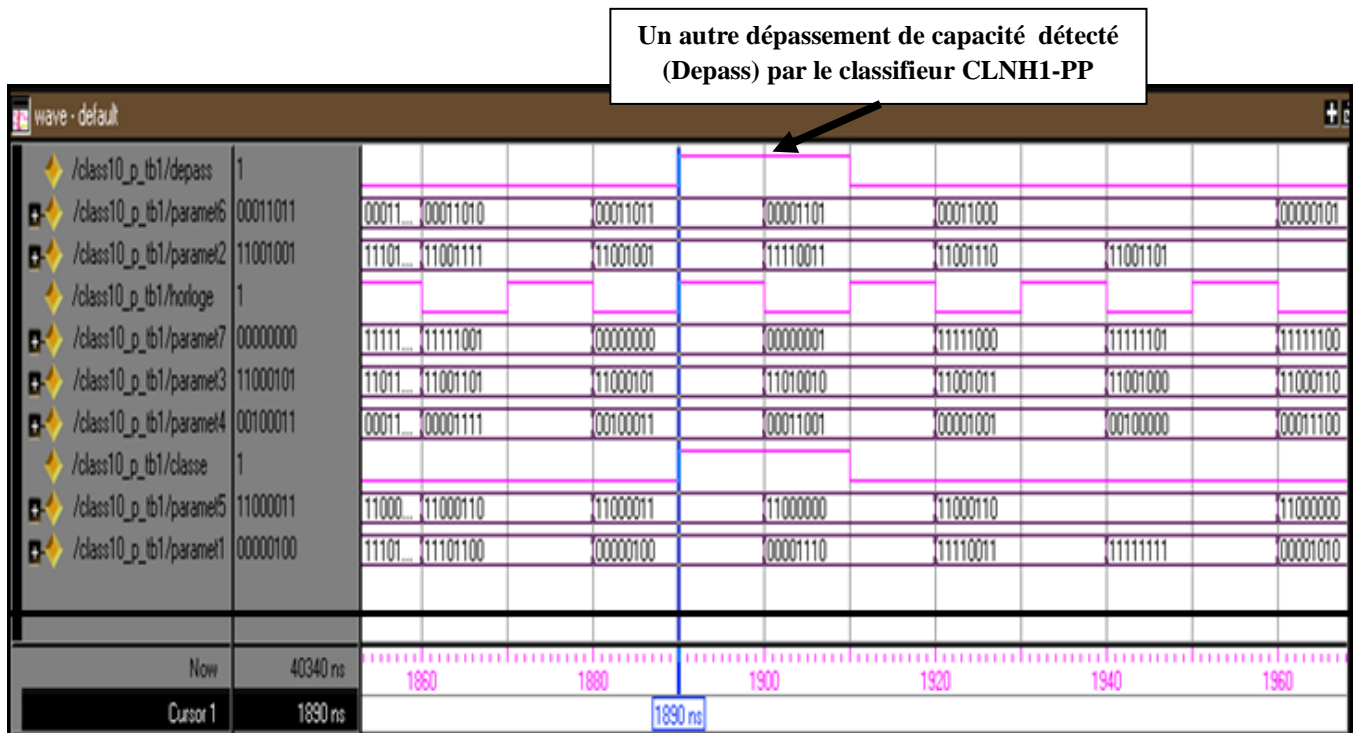


Figure 4.24. Le taux de classification obtenu des classifieurs CLN1 et CLNH1-PP.

Après la simulation du classifieur CLNH1-PP sous ModelSim, la figure 4.25 (a) montre l'élimination du dépassement qui a été produit auparavant par le classifieur CLNH-PP, mais nous constatons d'autres débordements de calcul (Figure 4.25 (b)) comme à l'instant 1890 ns.



(a)



(b)

**Figure 4.25. La simulation du classifieur CLNH1-PP sous ModelSim.**  
**(a). Le test des 6 premiers cycles cardiaques du sujet 106. (b). Le test des cycles cardiaques (92, 93, 94, 95, 96, 97, 98) du sujet 106.**

À partir des résultats obtenus, le classifieur CLNH1-PP a réalisé une mauvaise reconnaissance des extrasystoles ventriculaires (FN très élevé variant de 46 à 443) en introduisant un débordement important dans les calculs.

### 4.5.3. Le classifieur CLNH2-PP :

Puisque le classifieur CLNH1-PP n'a pas donné de meilleures performances et afin de remédier au problème du dépassement de capacité, nous avons utilisé le 3<sup>ème</sup> classifieur CLNH2-PP.

- **Résultats et discussion :**

Afin d'obtenir les résultats de l'implémentation des classifieurs CLN2 et CLNH2-PP (Tableau 4.12), nous avons divisé les paramètres d'entrée de la base de test par le coefficient  $K=4$ .



Tableau 4.12. Les performances des classifieurs CLN2 et CLNH2-PP.

Record	106		116		119		208		213		221	
K=4	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
NbR	1915	1949	2031	2161	1977	1976	2003	2102	2779	2753	2098	2184
TC (%)	94.94	96.63	85.30	90.76	100	99.95	78.15	82.01	97.71	96.80	86.87	90.43
TP	1454	1497	1925	2057	1535	1535	1018	1118	2615	2617	1707	1794
TN	461	452	106	104	442	441	985	984	164	136	391	390
FP	29	31	1	2	0	1	0	1	32	50	1	3
FN	0	0	29	25	0	0	245	165	4	3	139	78
OvF	/	0	/	0	/	0	/	0	/	0	/	0

Nous observons dans le tableau 4.12, une disparition totale des dépassements de capacité dans les différents sujets et une amélioration importante dans le taux de reconnaissance calculé sous l'environnement de ModelSim par rapport au taux calculé sous Matlab. Ainsi, nous remarquons que le nombre de battements réellement ESV reconnus comme de battements normaux, calculé par l'approche Software dans les sujets 208 et 221 est plus élevé à celui obtenu par l'approche Hardware (FN<sub>208</sub>=245 par rapport à FN<sub>H208</sub>=165, FN<sub>221</sub>=139 qui sont plus élevés de FN<sub>H221</sub>=78).

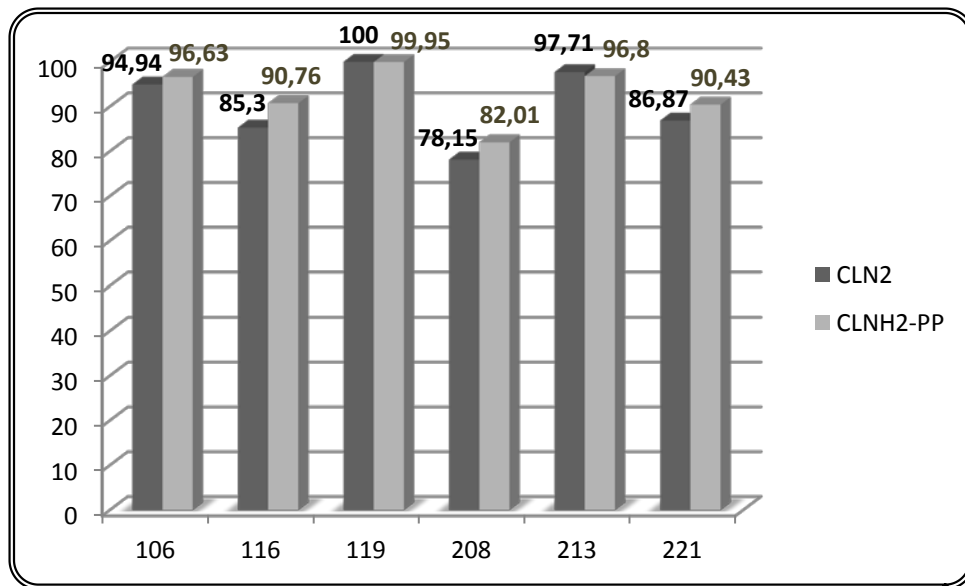


Figure 4.26. Le taux de classification obtenu des classifieurs CLN2 et CLNH2-PP.

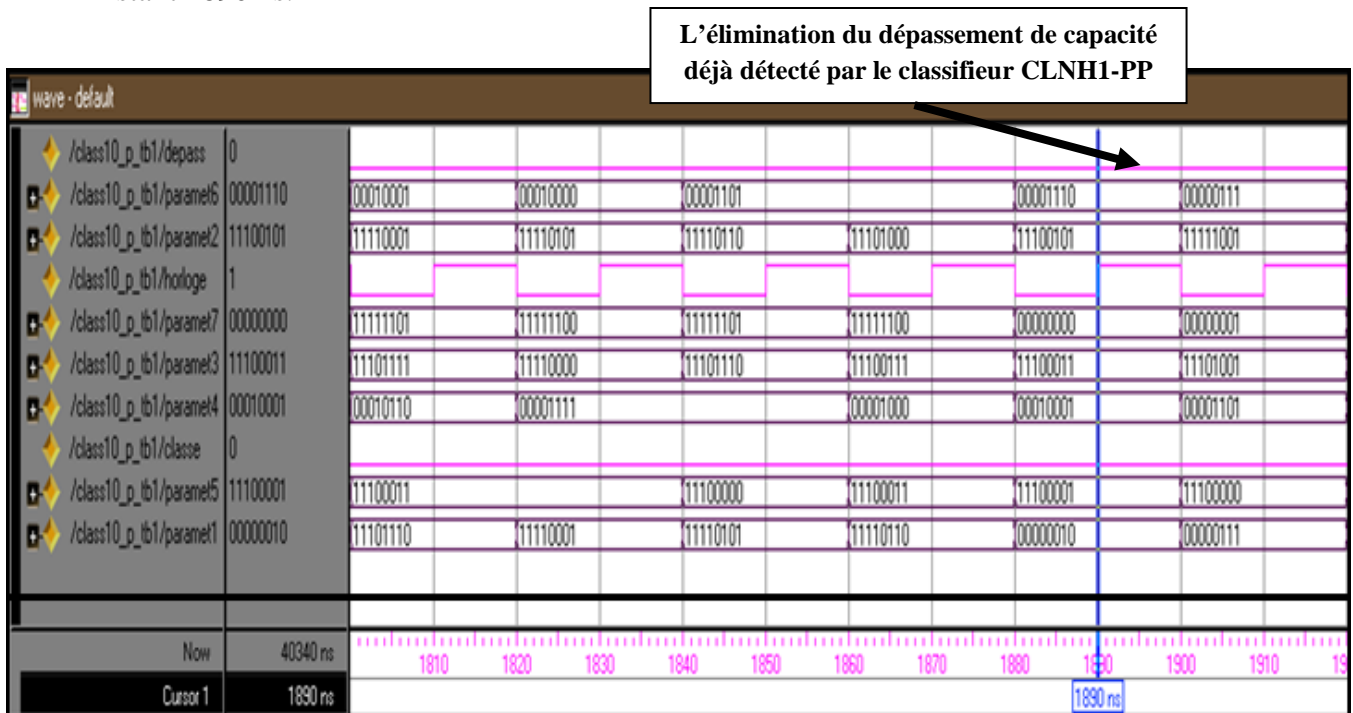
Le taux de reconnaissance obtenu par le classifieur CLN2 varie de 78.15% à 100% comparant avec les résultats donnés par le classifieur CLNH2-PP qui varie de 82.01% à 99.95% (Figure 4.26).

**Tableau 4.13. Le calcul de la sensibilité et de la spécificité des classifieurs CLN2 et CLNH2-PP.**

Record	106		116		119		208		213		221	
	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim	Matlab	Model Sim
Se (%)	100	100	98.51	98.79	100	100	80.60	87.14	99.84	99.88	92.47	95.83
Sp (%)	94.08	93.58	99.06	98.11	100	99.77	100	99.89	83.67	73.11	99.74	99.23

Ainsi, nous constatons sur le tableau 4.13 un taux de sensibilité, variant de 80.60% à 100% et un taux de spécificité, variant de 73.11% à 99.89% obtenus par l'approche Software comparant avec l'approche Hardware d'un taux de sensibilité, variant de 87.14% à 100% et d'un taux de spécificité, variant de 83.63% à 100%.

La simulation du classifieur CLNH2-PP sous ModelSim est présentée à la figure 4.27, en éliminant le dépassement de capacité détecté auparavant par le classifieur CLNH1-PP à l'instant 1890 ns.



**Figure 4.27. La simulation du classifieur CLNH2-PP sous l'environnement ModelSim.**

Lors de l'analyse des résultats obtenus dans les différentes implémentations, nous avons tiré quelques idées qui semblent intéressantes :

- ✚ Pour l'apprentissage de nos classifieurs (CLN, CLN1 et CLN2), nous avons utilisé une base de données qui contient 32658 battements dont le nombre de cas normaux (N) annotés par les médecins cardiologues est de 28766 et le nombre de cas ESV (V) est de 3892. Sachant que la représentativité d'une classe par rapport à l'autre peut biaiser l'apprentissage, nous avons choisi une base de données déséquilibrées afin de réaliser d'une part une mauvaise identification par l'approche Software des données positives (détection des FN) lors de la phase de test et d'essayer d'autre part l'amélioration de la reconnaissance de ces données par l'approche Hardware.

- ✚ Dans le Protocole2, nous avons choisi la valeur du coefficient K des puissances de deux afin de faciliter la conception matérielle de la division en remplaçant cette dernière par un décalage à droite. Notre but est de ne pas éliminer tous les dépassements de capacité (*Overflows*) mais de les réduire au maximum car le classifieur neuronal implémenté matériellement va servir d'aide au diagnostic pour le médecin, donc il faut aviser ce dernier si le calcul effectué est erroné (apparition des dépassements de capacité).
- ✚ En analysant les résultats obtenus, nous avons remarqué que l'implémentation matérielle du classifieur CLN2 soit dans la 1<sup>ère</sup> architecture (CLNH2-SP2) ou dans la 2<sup>ème</sup> conception (CLNH2-PP) donne globalement les mêmes performances pour les différents sujets de la base de test.

#### 4.6. Étude comparative entre les classifieurs CLNH-SP et CLNH-PP :

En adoptant les deux principes de calcul : **Semi-Parallèle (SP)** et **Parallèle-Parallèle (PP)**, nous avons pu réaliser deux architectures matérielles pour le classifieur neuronal CLN afin d'avoir des meilleures performances pour la reconnaissance des extrasystoles ventriculaires en temps réel. En effet, nous avons établi une étude comparative entre les deux architectures matérielles réalisées selon deux critères :

##### 4.6.1. Consommation des ressources FPGA :

Dans cette section, nous avons synthétisé les deux architectures matérielles réalisées sous l'environnement ISE Design 14.7, elles sont prévues d'être implémentées sur un FPGA du type **Spartan 3E**, qui intègre **1200K** portes logiques, cadencée à **50Mhz**. Les ressources requises pour l'implémentation matérielle du classifieur neuronal sont présentées dans le tableau suivant :

Tableau 4.14. Consommation des ressources matérielles.

Sommaire d'utilisation des composants (valeurs estimées)					
Ressources	Semi-Parallèle			Parallèle-Parallèle	
	Besoin	Utilisation	Disponible	Besoin	Utilisation
Nombre de tranches ( <i>Slices</i> )	615	7%	8672	594	6%
Nombre de bascules des tranches ( <i>Slice Flip Flops</i> )	82	< 1%	17344	0	0%
Nombre de LUTs à 4 entrées	1160	6%	17344	1141	6%
Nombre de blocs IOBs	13	5%	250	59	23%
Nombre de GCLKs	1	4%	24	1	4%
<b>La somme des ressources</b>	1871	<b>22%</b>	43634	1795	<b>39%</b>

Afin de clarifier les ressources matérielles disponibles dans le circuit FPGA utilisé, nous décrivons brièvement le *DataSheet* concernant le FPGA Spartan 3E de la société Xilinx :

Les blocs logiques configurables CLB (*Configurable Logic Blocs*) constituent les principales ressources logiques pour la mise en œuvre de circuits combinatoires. Chaque CLB contient quatre tranches (*Slices*), et chaque tranche contient deux générateurs de fonctions LUT (*Look Up Table*) et deux éléments de stockage (*Flip Flops*). Les LUT peuvent être utilisés comme des mémoires 16x1 (RAM16) ou comme des registres à décalage de 16 bits (SRL16). D'une manière générale, une conception logique est automatiquement mappée sur les tranches [SPARTAN\_2013].

Après l'étape de synthèse, nous constatons une diversité dans la consommation des ressources matérielles (Tableau 4.14) des deux conceptions réalisées dont nous trouvons une similarité dans l'utilisation des tranches, les générateurs de fonctions LUT à 4 entrées et le nombre de broches pour l'horloge principale GCLK (*Global Clock*).

Ainsi, nous remarquons que la deuxième conception utilisant le principe de calcul (PP) ne contient aucune bascule des blocs logiques car c'est une architecture matérielle purement combinatoire, mais nous observons une grande consommation des blocs d'entrée/sortie IOB (*Input/Output Blocks*) à cause du parallélisme effectué à l'entrée du classifieur neuronal.

Pour faire une comparaison sur la consommation des ressources matérielles entre les deux architectures réalisées, il faut soit calculer le nombre de portes logiques utilisé (1871 portes dans la 1<sup>ère</sup> conception devant 1795 portes dans la 2<sup>ème</sup> structure) si le cahier des charges exige la fabrication d'un circuit ASIC (*Application Specific Integrated Circuit*). Ou il faut prendre en considération le pourcentage d'utilisation des ressources matérielles (CLB, bascules, IOB,...) si l'implémentation de ces conceptions s'effectue sur un circuit FPGA. Alors d'après le tableau 4.14, le classifieur CLNH-SP2 utilise 22% des ressources matérielles du FPGA Spartan 3E tandis que le deuxième classifieur CLNH-PP utilise 39% de ces ressources, donc nous concluons que ce dernier est le plus consommateur en ressources vue de l'exploitation du parallélisme au maximum à tous les niveaux.

Enfin, nous avons réalisé les deux conceptions matérielles en partant généralement du niveau fonctionnel jusqu'au niveau transfert de registre RTL (*Register Transfert Logic*). Ce dernier est un niveau d'estimation qui se base sur la description de l'implémentation sous forme de registres et de bascules. Alors, les figures 4.28 et 4.29 présentent les structures RTL générées par ISE Design 14.7 des deux architectures matérielles.

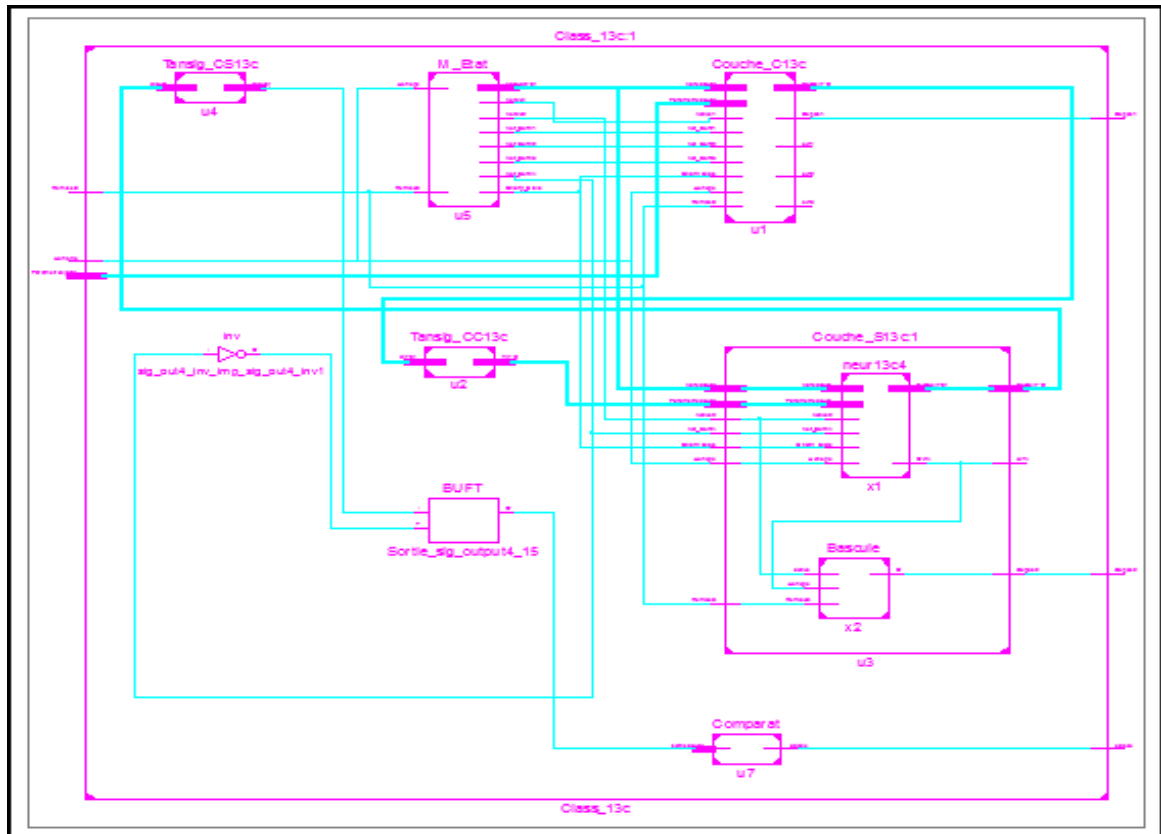


Figure 4.28. La structure RTL du classifieur CLNH-SP2.

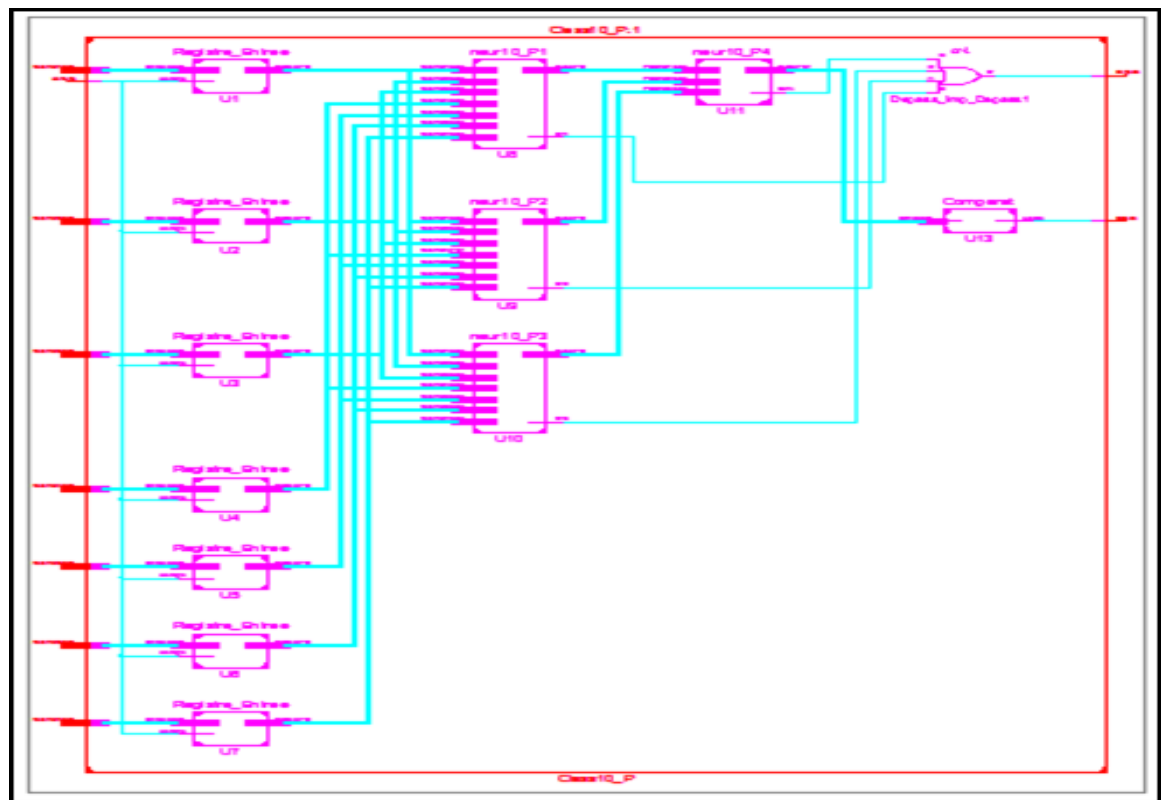


Figure 4.29. La structure RTL du classifieur CLNH-PP.

#### 4.6.2. Temps de calcul :

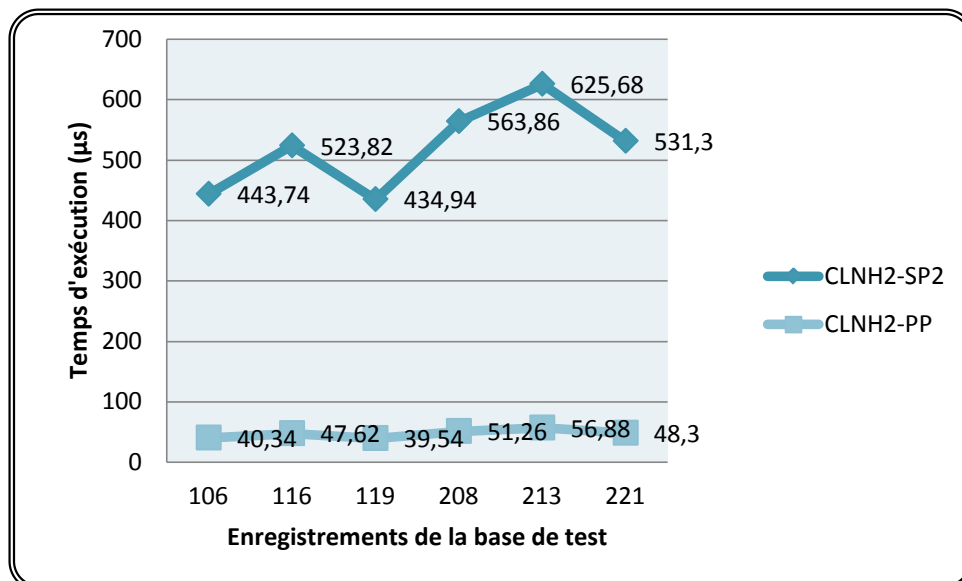
Dans cette étude, nous avons présenté le classifieur CLN2 ainsi que leurs implémentations matérielles (CLNH2-SP2 et CLNH2-PP) car il ne provoque aucun dépassement de capacité pour les deux conceptions matérielles et pour les différents sujets de la base de test (un bon classifieur avec des résultats corrects).

Le tableau 4.15 récapitule le temps de calcul pour la reconnaissance des extrasystoles ventriculaires dans les différents enregistrements de la base de test. Nous présentons le temps effectué par les deux conceptions réalisées sous l'environnement ModelSim et à titre indicatif, nous avons exposé le temps de calcul sous Matlab, ce dernier fonctionne sous environnement Windows qui est un système multitâche.

**Tableau 4.15. Le temps d'exécution des classifieurs CLNH2-SP2 et CLNH2-PP.**

Sujets	Taille	Matlab		ModelSim			
				CLNH2-SP2		CLNH2-PP	
		Taux de classification (%)	Temps d'exécution ( $\mu$ s)	Taux de classification (%)	Temps d'exécution ( $\mu$ s)	Taux de classification (%)	Temps d'exécution ( $\mu$ s)
106	2017	94.94	58346	96.68	443.74	96.63	40.34
116	2381	85.30	65927	90.80	523.82	90.76	47.62
119	1977	100	67150	100	434.94	99.95	39.54
208	2563	78.15	69594	82.05	563.86	82.01	51.26
213	2844	97.71	59958	96.84	625.68	96.80	56.88
221	2415	86.87	60250	90.48	531.30	90.43	48.30

En réalisant une architecture purement parallèle, nous avons abouti à un temps d'exécution très faible devant la conception qui utilise le Semi-Parallèle (Figure 4.30).



**Figure 4.30. Le temps d'exécution des classifieurs CLNH2-SP2 et CLNH2-PP.**

D'une façon générale, le temps d'exécution pour reconnaître une classe en utilisant le Semi-Parallèle est après 10 cycles d'horloge mais en adoptant le principe Parallèle-Parallèle, la classe est spécifiée dans le 1<sup>er</sup> cycle d'horloge. Cette dernière architecture CLNH-PP est totalement combinatoire mais nous utilisons l'horloge afin de synchroniser les résultats à la sortie du classifieur.

#### **4.7. Conclusion :**

Dans ce chapitre, nous avons présenté les deux approches d'implémentation d'un classifieur neuronal pour la reconnaissance des extrasystoles ventriculaires, nous avons choisi le classifieur de type perceptron multicouche car il présente l'avantage d'être un outil puissant tout en restant facile à manipuler et en raison de son architecture hautement parallélisable. Ces avantages sont faits de lui un parfait candidat pour une implantation matérielle.

En implémentant le classifieur neuronal matériellement selon deux principes de calcul différents (SP, PP), la structure est figée puisque les valeurs des poids synaptiques et des biais sont obtenues par l'approche software (sous l'environnement Matlab) à partir d'une base exhaustive d'enregistrements cardiaques (MIT-BIH). Nous avons étudié les deux approches en présentant leurs performances dans la classification des extrasystoles ventriculaires.

En maintenant la représentation des données sur un octet dans la conception matérielle, nous étions confrontés au problème du dépassement de capacité qui a été résolu en adoptant deux protocoles de calcul expérimentaux (Protocole1, Protocole2). Ces derniers ont été utilisés afin de réduire au maximum le débordement en avisant le médecin lors de son diagnostic.

En examinant les performances obtenues par les deux approches, nous constatons que les deux implémentations matérielles réalisées ont apporté des améliorations considérables lors de la reconnaissance des extrasystoles ventriculaires, et particulièrement en termes de temps, en éliminant tout risque de débordements dans le calcul.

Les approches développées dans le cadre de cette thèse concernent l'implantation des réseaux de neurones artificiels, dans notre cas d'application le perceptron multicouche en phase de classification, sur un circuit programmable du type FPGA en vue d'une éventuelle exploitation dans un système embarqué qui sera défini dans le prochain chapitre.



*Chapitre 5*

*Classification Neuronale des  
Arythmies Cardiaques à Base  
de FPGA.*



## 5.1. Introduction :

Lors de cette thèse, nous avons réalisé un classifieur neuronal du type multicouche (MLP) sur un circuit FPGA. Cette implantation est effectuée en vue d'une future exploitation dans un système embarqué capable de détecter les arythmies cardiaques en temps réel.

Ce chapitre portera sur l'étude d'un modèle de classification des données médicales intégrant le classifieur neuronal déjà présenté dans le chapitre précédent. Nous avons même proposé une phase de correction lors de la détection des dépassements de capacité. Il s'agira ici de fiabiliser la reconnaissance des arythmies cardiaques dans le but de réaliser un système automatisé d'alerte des anomalies cardiaques.

Nous présentons la conception du système matériel pour la reconnaissance en particulier des extrasystoles ventriculaires en temps réel, nous montrons les différentes configurations du modèle, ainsi que les techniques distinctes de conception. Nous détaillons plus, la dernière technique puisqu'elle fait l'objet de nos travaux. Ensuite, nous exposons les performances de nos modèles avec ceux décrit dans l'état de l'art. Enfin, nous terminons ce chapitre par une conclusion.

## 5.2. La conception d'un modèle de classification des arythmies cardiaques :

Dans ce travail, nous avons proposé deux architectures matérielles (**CLNH-SP** et **CLNH-PP**) permettant de mettre en œuvre deux modèles de classification des extrasystoles ventriculaires (**SYST-SP** et **SYST-PP**) sur cible FPGA. Les deux architectures réalisées ont montré un compromis entre le gain effectué en temps de classification et les performances attendues.

La conception de ces deux modèles (**SYST-SP** et **SYST-PP**) utilise une méthode identique afin de réaliser une application originale complète de classification. Nous présenterons dans cette section seulement l'étude du premier modèle **SYST-SP**, la conception de ce dernier est basée sur la collaboration des quatre classifieurs neuronaux (**CLNH-SP2**, **CLNH1-SP2**, **CLNH2-SP2** et **CLNH3-SP2**) déjà exposés dans le chapitre 4.

Dans cette étude, notre modèle de classification **SYST-SP** est créé à partir de quatre configurations différentes (A, B, C et D) (Figure 5.1).

- ✚ La 1<sup>ère</sup> configuration (A) concerne le classifieur neuronal **CLNH-SP2**, qui utilise les valeurs des paramètres d'entrée du réseau neuronal **sans division**.
- ✚ La 2<sup>ème</sup> configuration (B) représente le classifieur **CLNH1-SP2**, qui utilise les valeurs des paramètres d'entrée divisées par deux ( $/2$ ). Pour ce faire, nous ajoutons à l'entrée du réseau un registre de décalage à droite qui permet de réaliser cette division. Dans cette configuration, il y aura un changement respectif des valeurs des poids

synaptiques stockées dans la ROM, de biais mémorisé dans le registre, correspondants à chaque Neurone\_NC, ainsi que le contenu des blocs de la fonction d'activation.

- ✚ La 3<sup>ème</sup> configuration (C) décrit le classifieur CLNH2-SP2, utilisant un registre de décalage à droite (un décalage de 2 bits successifs) permettant la division des valeurs des paramètres d'entrée par quatre (/4). Dans cette configuration, il y aura un autre changement dans le contenu des blocs Poids\_Rom, Registre\_Biais et Fonct\_Tansig du classifieur neuronal.
- ✚ La 4<sup>ème</sup> configuration (D) concerne le classifieur CLNH3-SP2 qui utilise les valeurs des paramètres d'entrée divisées par huit (/8). En utilisant ce classifieur, nous remarquons qu'il n'y a pas un débordement dans le calcul, il est rarement utilisé car nous avons constaté dans le chapitre précédent que les dépassements de capacité sont éliminés totalement dans le 2<sup>ème</sup> ou le 3<sup>ème</sup> classifieur du modèle. Le même changement sera adopté comme les configurations précédentes.

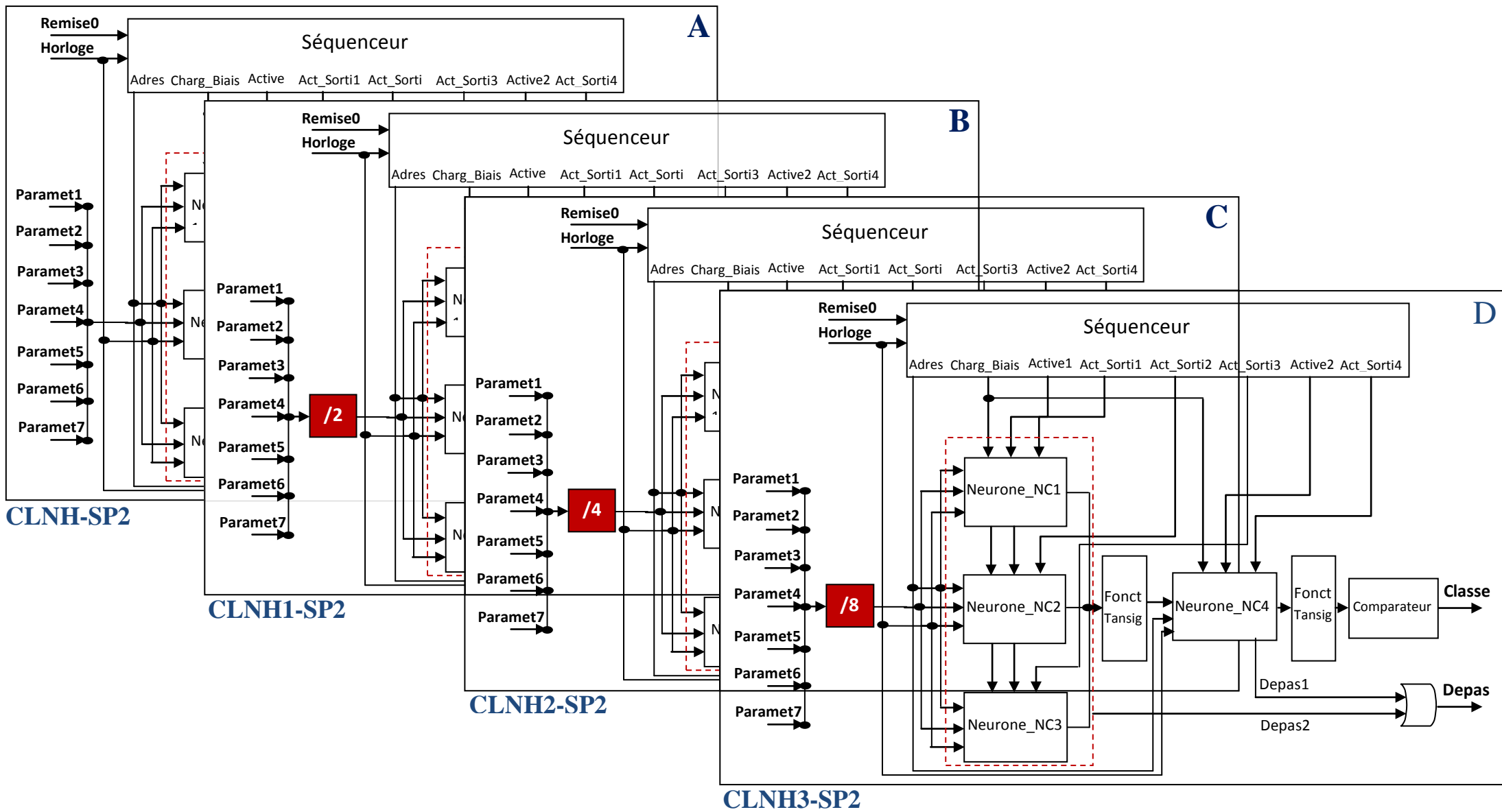
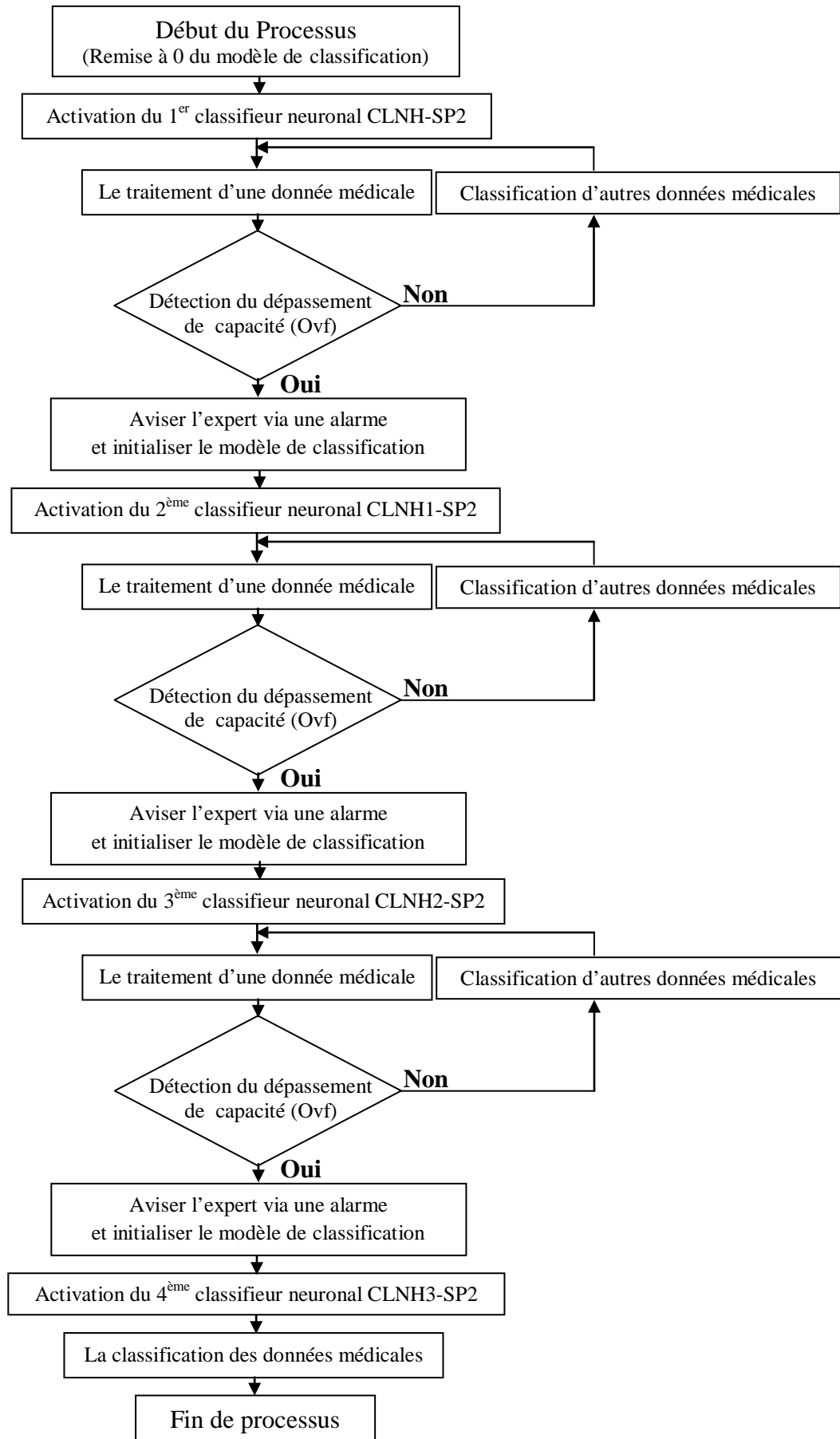


Figure 5.1. Les quatre configurations du modèle de classification SYST-SP.

L'idée principale de notre approche consiste à appliquer une chaîne de traitement (Organigramme 5.1) afin de réaliser le modèle de classification des arythmies cardiaques.



### Organigramme 5.1. Le fonctionnement du modèle de classification SYST-SP.

À la détection du dépassement de capacité, il faut aviser l'expert par un système d'alerte afin de ne pas prendre le résultat en considération. Une alerte consiste à allumer une lampe, par exemple, indiquant que le calcul effectué est erroné.

Afin de réaliser le modèle de classification, nous avons proposé différentes techniques de liaisons entre les classifieurs neuronaux (CLNH-SP2, CLNH1-SP2, CLNH2-SP2 et CLNH3-SP2) :

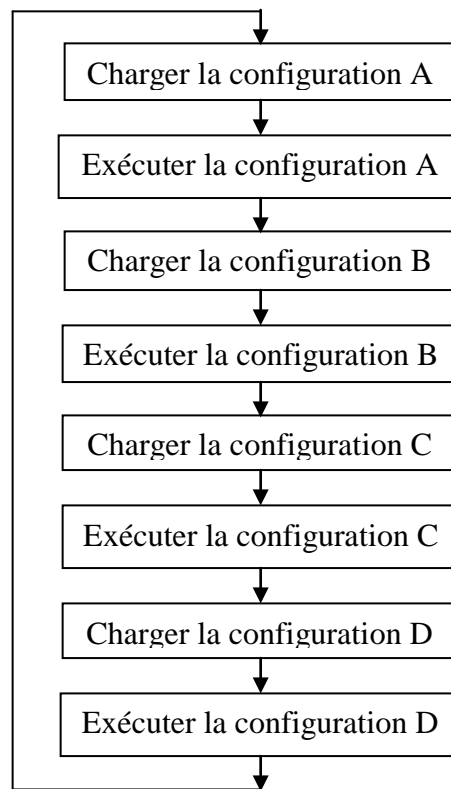
- La reconfiguration dynamique globale.
- La reconfiguration dynamique partielle.
- L'architecture en électronique câblée.

Aujourd'hui, les fabricants de FPGA mettent à la disposition des utilisateurs les dernières technologies disponibles, proposant des performances et des capacités permettant l'implémentation de la plupart des applications courantes. Par conséquent, les FPGA peuvent actuellement être reconfigurables par d'autres architectures au moment de leurs exécutions comme la reconfiguration **dynamique globale** et **dynamique partielle**. Ces dernières possibilités augmentent la flexibilité et les possibilités des FPGA, mais leurs réalisations sont faites par des techniques complexes. Dans cette section, nous ne présentons que des notions sur les différentes possibilités de reconfiguration, vu que leur accomplissement serait une tâche difficile mais elles peuvent être étudiées lors des travaux futurs. Notre étude actuelle concerne l'architecture embarquée en électronique câblée seulement.

#### 5.2.1. La reconfiguration dynamique globale :

Le circuit FPGA reconfigurable dynamiquement possède un certain nombre d'interfaces de configuration qui sont utilisés pour télécharger les données de configuration (le fichier de configuration) afin de pouvoir changer à tout moment son fonctionnement. La reconfiguration dynamique permet à tous les bits du fichier de configuration (*Bistream*) d'être réécrit même s'ils n'ont pas changé entre deux configurations, ce qui est évidemment une perte de temps.

La reconfiguration dynamique de notre modèle de classification est présentée par l'organigramme suivant :



**Organigramme 5.2. La reconfiguration dynamique du modèle SYST-SP.**

### 5.2.2. La configuration dynamique partielle du FPGA :

Contrairement à la reconfiguration dynamique, nous pouvons configurer partiellement le FPGA afin de n'utiliser que les ressources nécessaires à l'implémentation de l'application. Ce type de reconfiguration permet d'économiser la consommation en énergie et en ressources logiques du FPGA, ainsi que le temps de reconfiguration.

La reconfiguration partielle est un processus de changement d'une zone en FPGA sans altérer les autres applications, son flux est divisé en deux régions:

- La région statique : qui sera composée de toute la partie fixe de l'architecture matérielle, elle est programmée au démarrage et ne changera jamais.
- La région dynamique : est une partie de l'architecture reconfigurée plusieurs fois avec des modules différents en cours d'exécution de l'application sur le FPGA.

La reconfiguration partielle permet de créer deux fichiers de configuration, l'un pour la région statique (fixe) et l'autre pour la région dynamique (variable).

#### **Remarque :**

L'inconvénient majeur d'une reconfiguration dynamique (globale ou partielle) est le temps d'exécution de cette technique, qui est considéré important par rapport à une configuration

simple d'une application (de quelques ms pour passer d'une configuration à l'autre), et cela introduit un impact sur les performances d'un système en temps réel, notamment si la modification intervient fréquemment. En effet, l'utilisation de cette méthode dans notre approche est déconseillée vu que nous avons plusieurs architectures à configurer.

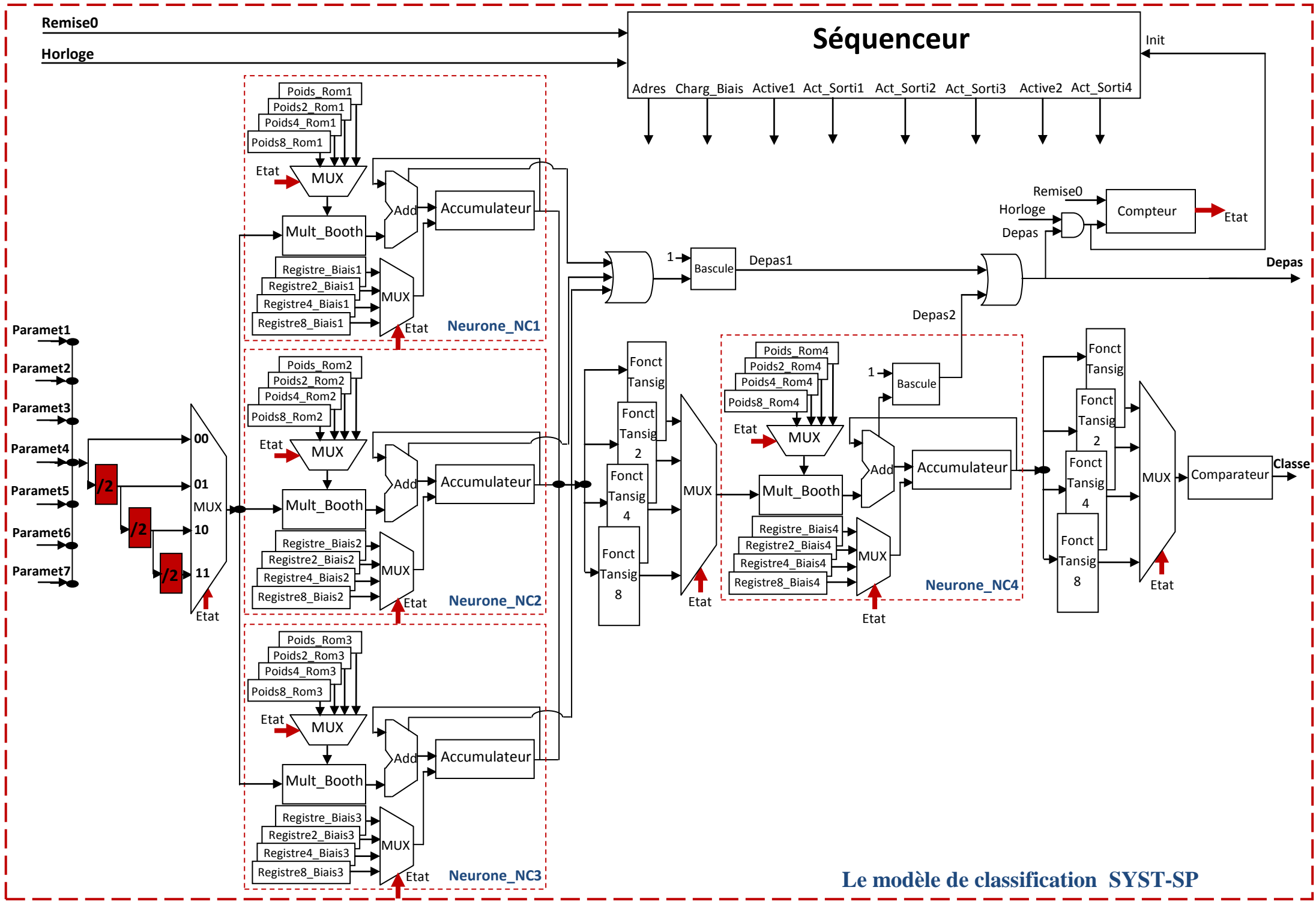
### **5.2.3. Architecture en électronique câblée :**

#### **5.2.3.1. L'architecture matérielle du modèle SYST-SP :**

Pour cette technique, la liaison entre les classifieurs neuronaux se fait par l'ajout de leurs blocs dans l'architecture matérielle du classifieur neuronal CLNH-SP2 (décrit dans le chapitre 3). Les blocs à ajouter sont les blocs qui changent de contenu d'un classifieur à l'autre. En effet, l'architecture matérielle du SYST-SP (Figure 5.2) est décrite comme suit :

- Chaque Neurone\_NC contient :
  - Les blocs changeants de contenu comme les ROM (Poids\_Rom), les registres (Registre\_Biais). La permutation entre les blocs ROM (Poids\_Rom, Poids2\_Rom, Poids4\_Rom et Poids8\_Rom) des quatre classifieurs s'exécute à l'aide d'un multiplexeur (4 vers 1) et le même principe est utilisé pour les registres (Registre\_Biais, Registre2\_Biais, Registre4\_Biais et Registre8\_Biais).
  - Les blocs fixes sont le multiplicateur et l'accumulateur.
- Le modèle comporte lui aussi :
  - Les blocs changeant de contenu sont les fonctions d'activation de chaque couche du réseau (Fonct\_Tansig, Fonct2\_Tansig, Fonct4\_Tansig, et Fonct8\_Tansig), le changement entre ces blocs s'effectue par un multiplexeur (4 vers 1).

Les blocs fixes sont les quatre blocs Neurone\_NC, le séquenceur, le comparateur, un compteur de 4 états et trois registres de décalage à droite qui se pose à l'entrée du réseau permettant la division des paramètres d'entrée par 2, 4 et 8 respectivement. Le choix entre ces derniers est procédé par un multiplexeur (4 vers 1) [BABA\_06/2014].





### Figure 5.2. L'architecture matérielle du modèle SYST-SP.

Le fonctionnement du modèle SYST-SP se déroulera comme suit :

- Au début de la phase de classification, le compteur est initialisé à l'état (00).
- Tous les multiplexeurs du système s'aiguillent vers la 1ère position (00). Alors le système utilise les blocs du 1<sup>er</sup> classifieur matériel CLNH-SP2.
- Si le classifieur CLNH-SP2 détecte un dépassement de capacité, le compteur est incrémenté à l'état 01, en initialisant le séquenceur par le signal (Init), cela permet d'aiguiller tous les multiplexeurs vers la position (01). Alors, notre système utilise les blocs du 2<sup>ème</sup> classifieur CLNH1-SP2.
- Si un dépassement de capacité est repéré lors de la classification des données médicales par CLNH1-SP2, le compteur est incrémenté à l'état (10) en initialisant toujours le séquenceur, ce qui permet d'aiguiller tous les multiplexeurs vers la 3<sup>ème</sup> position. Ainsi, le système utilise les blocs du classifieur suivant CLNH2-SP2.

Le même principe est utilisé avec le prochain classifieur CLNH3-SP2 si le précédent détecte un débordement dans le calcul.

#### Remarque :

En résumé, le compteur est incrémenté à chaque détection de dépassement de capacité, permettant au modèle SYST-SP de sélectionner un des classifieurs neuronaux pour la classification des extrasystoles ventriculaires en temps réel. En conséquence, l'architecture en électronique câblé possède un temps d'exécution très faible (en ns) par rapport aux méthodes décrites auparavant, puisque c'est une configuration simple d'une architecture matérielle sur la cible FPGA.

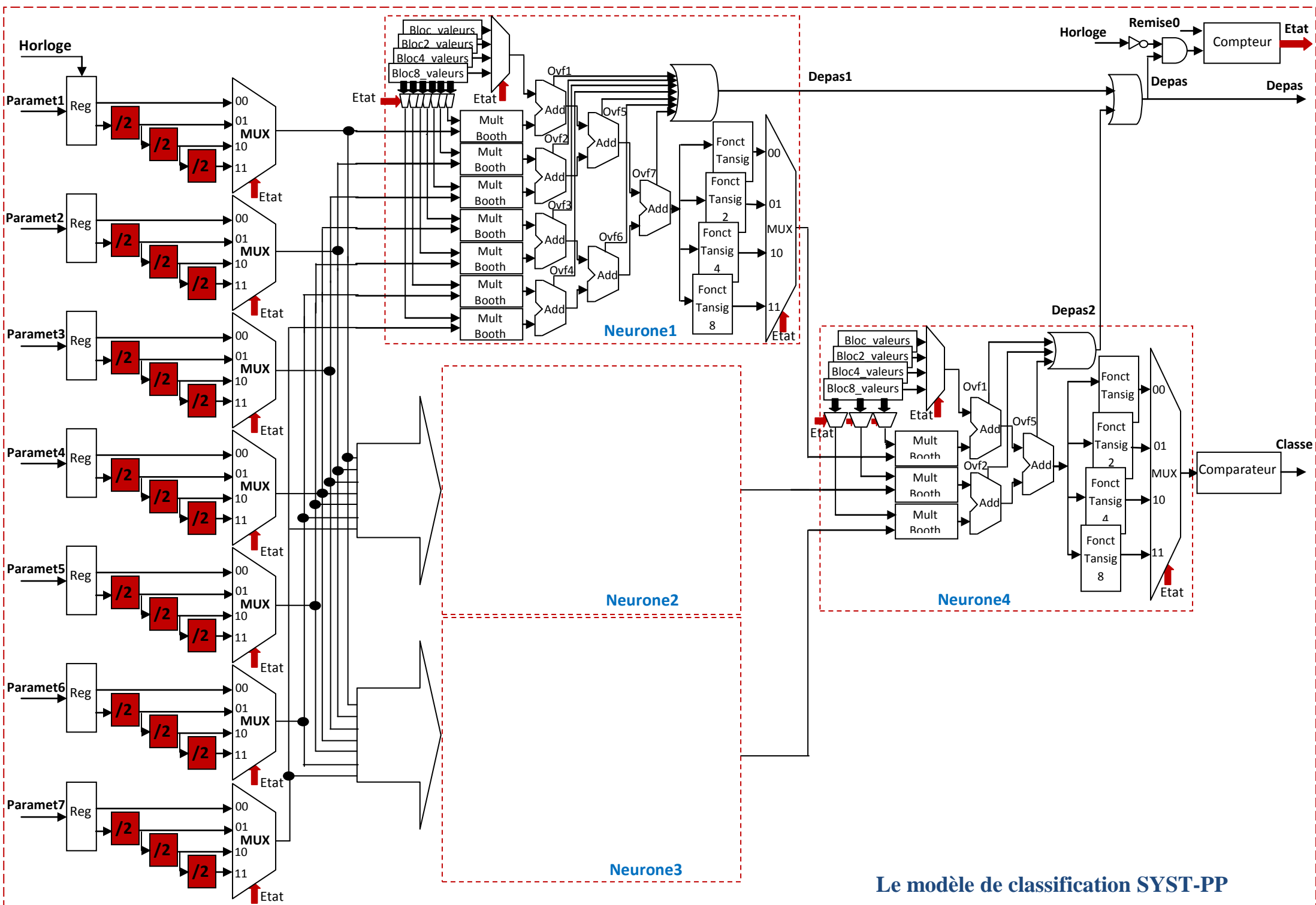
#### 5.2.3.2. L'architecture matérielle du modèle SYST-PP :

Nous présentons dans la figure suivante l'architecture matérielle de notre deuxième modèle SYST-PP de classification des extrasystoles ventriculaires, cette conception est basée sur l'ajout des différents blocs qui changent de contenu d'un classifieur à un autre (CLNH1-PP, CLNH2-PP, CLNH3-PP) dans l'architecture matérielle du classifieur CLNH-PP (décrit dans le chapitre 3). Alors, l'architecture matérielle du SYST-PP (Figure 5.3) est décrite comme suit :

- Chaque Neurone contient :
  - Les blocs changeant de contenu comme les blocs qui contiennent les valeurs des poids synaptiques et du biais de chaque classifieur (Bloc\_Valeurs, Bloc2\_Valeurs, Bloc4\_Valeurs, Bloc8\_Valeurs), ainsi, les fonctions d'activation (Fonct\_Tansig, Fonct2\_Tansig, Fonct4\_Tansig, Fonct8\_Tansig). Le choix entre ces blocs qui changent de contenu s'effectue par des multiplexeurs (4 vers 1).
  - Les blocs fixes sont les multiplicateurs et les additionneurs.

- Le système comporte que des blocs fixes, ils soient les quatre blocs Neurone, le comparateur, un compteur de 4 états et normalement 21 registres de décalage à droite (3 registres pour chaque paramètre d'entrée du réseau permettant la division du paramètre par 2, 4 et 8 respectivement). Le choix entre ces derniers est procédé par un multiplexeur (4 vers 1).

Le même fonctionnement du SYST-SP est appliqué sur le SYST-PP pour une reconnaissance des extrasystoles ventriculaires avec élimination totale de dépassement de capacité.



**Le modèle de classification SYST-PP**

Figure 5.3. L'architecture matérielle du modèle SYST-PP.

## 5.2.3.3. La synthèse des deux modèles de classification :

Après la synthèse des deux modèles de classifications SYST-SP et SYST-PP sous l'environnement de ISE Design 14.7 de Xilinx, nous remarquons une augmentation considérable dans l'utilisation des ressources matérielles du circuit FPGA Spartan 3E par rapport aux classifieurs neuronaux (CLNH-SP et CLNH-PP) . Donc d'après le tableau 5.2, le modèle SYST-SP utilise 45% des ressources matérielles, tandis que le deuxième modèle SYST-PP consomme 85% de ces ressources.

Tableau 5.2. Consommation des ressources matérielles des deux modèles de classification.

Sommaire d'utilisation des composants (valeurs estimées)					
Ressources	SYST-SP			SYST-PP	
	Besoin	Utilisation	Disponible	Besoin	Utilisation
Nombre de tranches ( <i>Slices</i> )	805	9%	8672	2556	29%
Nombre de bascules des tranches ( <i>Slice Flip Flops</i> )	514	2%	17344	128	0%
Nombre de LUTs à 4 entrées	1431	8%	17344	4900	28%
Nombre de blocs IOBs	17	6%	250	60	24%
Nombre de GCLKs	5	20%	24	1	4%
La somme des ressources	2772	45%	43634	7645	85%

## 5.3. Travaux connexes :

Cette thèse présente l'implémentation matérielle d'un modèle de classification des extrasystoles ventriculaires sur un circuit programmable du type FPGA, le modèle est capable

- de classer les données médicales en temps réel,
- de détecter les erreurs de calcul (*Overflow*) pendant la phase de classification.
- de sélectionner entre les classifieurs afin d'éliminer le risque d'erreur.

Le système est constitué par une combinaison de quatre classifieurs de type perceptron multicouche, nous avons utilisé l'algorithme Pan-Tompkins pour la détection du complexes QRS [PAN\_1985] et une base d'apprentissage constituée de 14 enregistrements de la base de données MIT-BIH, et pour la phase de test, nous avons choisi 6 enregistrements différents de la base d'apprentissage.

Les performances réalisées par les deux classifieurs sont :

- Le classifieur à base de SYST-SP réalise un taux de classification moyen de **92.81%** (varie de 82.05% à 100%), une sensibilité moyenne de **96.94%** (varie de 87.14% à 100%) et une spécificité moyenne de **94.33%** (varie de 73.65% à 100%) avec une **élimination totale des overflows**.

- Le classifieur à base de SYST-PP réalise un taux de classification moyen de **92.76%** (varie de 82.01% à 99.95%), une sensibilité moyenne de **96.94%** (varie de 87.14% à 100%) et une spécificité moyenne de **94.95%** (varie de 73.11% à 100%) **sans risque de dépassement de capacité.**

Dans cette section, nous exposons nos modèles de classification réalisés avec la plupart de ceux décrit dans l'état de l'art (chapitre 1), ayant utilisé l'implémentation matérielle d'un perceptron multicouche et ayant validé leurs méthodes de classification sur la base de données MIT-BIH. Malgré la différence entre le nombre de classes à détecter, nous présentons seulement les performances obtenues des cas normaux et des cas ESV (extrasystoles ventriculaires).

La base de données MIT-BIH [MIT\_2010] comprend les arythmies suivantes :

- N : Battement normal (*Normal beat*).
- L: Battement de bloc de branche gauche (*Left bundle branch block beat*).
- R: Battement de bloc de branche droite (*Right bundle branch block beat*).
- A: Battement auriculaire prématuré (*Atrial premature beat*).
- S: Battement supraventriculaire prématuré (*Supraventricular premature beat*).
- V : Contraction ventriculaire prématurée (*Premature ventricular contraction*).
- F: Fusion de rythme ventriculaire et normal (*Fusion of ventricular and normal beat*).
- f: Fusion de rythme régulier et normal (*Fusion of paced and normal beat*).
- Q: Battement inclassable (*Unclassifiable beat*).
- P: Rythme stimulé (*Paced rhythm*).

Jewajinda et Chongstitvatana, ont proposé dans leur papier [JEWAJINDA\_2010] un modèle combinant les fonctions Hermite, pour la caractérisation des battements ECG, et un classifieur neuronal basé sur un algorithme génétique parallèle du type cellulaire (EC-CGA) approprié pour le matériel. L'apprentissage du classifieur a été effectué sur une base de données composée de 22 enregistrements de la base MIT-BIH, et pour la phase de test, elle a été appliquée sur une base de données de 22 enregistrements différents de la première base. Ce classifieur permet de détecter cinq classes (N, S, V, F et Q) en le mettant en œuvre sur un circuit reconfigurable du type FPGA. La performance de l'implémentation pour la détection des extrasystoles ventriculaires a été évaluée en termes de taux de classification, de sensibilité et de spécificité respectivement **97.7%**, **85.8%** et **98.7%**.

Ozdemir et Danisman [OZDEMIR\_2011], ont présenté une architecture totalement parallèle d'un perceptron multicouche implémentée sur un circuit FPGA du type Cyclone III EP3C120F780C7 d'Altera, en utilisant une représentation de données à virgule flottante. Les caractéristiques du signal ECG sont réduites par l'analyse des composantes principales (ACP), ils ont construit une base de 4 enregistrements (205, 208, 210 et 213) de la base de données MIT-BIH dont le 2/3 était utilisé pour l'apprentissage du classifieur et le reste pour le test. Le classifieur est capable de classer 3 battements d'ECG (N, V et F) avec un taux de classification de **97.66%**.

Dans le papier de Jatmiko et al [JATMIKO\_2011], ils ont utilisé la transformation par ondelettes discrètes (DWT) pour caractériser les battements ECG, et un classifieur neuronal basé sur l'algorithme de rétro-propagation afin de classer ces battements en classe normale et anormale. Le système a été implémenté sur un FPGA du type Spartan-3A DSP XC3SD3400A de Xilinx en réalisant un taux de classification qui varie de **90%** à **100%**. Le même modèle a été proposé par Egila et al [EGILA\_2016], en ajoutant à son entrée un filtre (FIR) pour filtrer le bruit du signal ECG, ils ont utilisé une base d'apprentissage de 48 enregistrements d'ECG normaux et 42 enregistrements anormaux. Pour la base de test, 24 enregistrements normaux et 21 anormaux. Leur classifieur a été réalisé sur un FPGA en utilisant la carte XILINX Spartan 3AN-XC3S700AN. Les performances obtenues ont été présentées avec un taux de **95.23%** pour l'identification des battements anormaux, et un taux de classification globale de **97.8%**.

Une implémentation d'un modèle de classification sur un FPGA du type Virtex 7 de Xilinx a été proposé par Travieso-González et al [GONZALEZ\_2013]. Ce modèle est basé sur les techniques de la transformée d'ondelettes numériques (DWT), de l'analyse de composantes principales (ACP) et un perceptron multicouche basé sur l'algorithme de rétro-propagation, capable de classer 8 battements d'ECG (N, L, R, P, V, A, F, f). L'apprentissage du classifieur a été effectué sur la moitié d'une base construite de la base de données MIT-BIH, et le test sur le reste de cette base. La performance de l'implémentation pour la détection des extrasystoles ventriculaires a été évaluée en termes de taux de classification, de sensibilité et de spécificité respectivement **98.25%**, **97.90%** et **100%**.

Tableau 5.2. Présentation des différentes méthodologies mises en œuvre sur FPGA.

Auteurs	Méthodes	Nombre de classe	Taux de classification	Sensibilité	Spécificité	Le FPGA utilisé	Le nombre de ressources logiques	Temps d'exécution
Jewajinda et Chongstitvatana [JEWAJINDA_2010].	Hermite +Neuronal Network	5	97.7%	85.8%	97.7%	ND	ND	ND
Ozdemir et Danisman [OZDEMIR_2011]	ACP+ Neuronal Network	3	97.66%	ND	ND	Cyclone III EP3C120F780C7 d'Altera	ND	ND
Jatmiko et al [JATMIKO_2011]	DWT+ Neuronal Network	2	varie de 90% à 100%	ND	ND	Spartan 3A DSP XC3SD3400A de Xilinx	23805	ND
Egila et al [EGILA_2016].	FIR+DWT+ Neuronal Network	2	97.8%	95.23% cas anormaux.	100 % cas normaux.	Spartan 3AN XC3S700AN de Xilinx	12311	ND
Travieso-González et al [GONZALEZ_2013]	DWT+ACP+ Neuronal Network	8	98.25%	97.90%	100%	Virtex 7 de Xilinx	84,7%	158,23 ns
Notre 1 <sup>er</sup> modèle SYST-SP	Pan-Tompkins+ Neuronal Network	2	<b>92.81%</b> (varie de 82.05% à 100%)	<b>96.94%</b> (varie de 87.14% à 100%)	<b>94.33%</b> (varie de 73.65% à 100%)	Spartan 3E XC1200K de Xilinx	45%	200 ns
Notre 2 <sup>ème</sup> modèle SYST-PP	Pan-Tompkins+ Neuronal Network	2	<b>92.76%</b> (varie de 82.01% à 99.95%)	<b>96.94%</b> (varie de 87.14% à 100%)	<b>94.95%</b> (varie de 73.11% à 100%)	Spartan 3E XC1200K de Xilinx	85%	20 ns

D'après le tableau 5.2, nous remarquons une diversité dans les performances des classifieurs matériels, normalement cela est dû à la dissemblance de la base d'apprentissage ainsi que la base de test, comme dans le travail [OZDEMIR\_2011] dont ils ont premièrement équilibré leur base de données, puis ils l'ont divisé en 2/3 pour l'apprentissage du classifieur et le reste pour le test. Le même cas pour [GONZALEZ\_2013] où ils ont utilisé la moitié de leur base de données pour l'apprentissage et l'autre moitié pour le test, mais par rapport à nos modèles de classification, nous avons testé avec une base totalement différente de la base d'apprentissage, en sachant ainsi que cette dernière était déséquilibrée. En revanche, nos deux modèles de classification SYST-SP et SYST-PP ont réalisé une bonne reconnaissance des extrasystoles ventriculaires en temps réel.

#### **5.4. Conclusion :**

Dans ce chapitre, nous avons présenté les concepts fondamentaux d'un modèle de classification des extrasystoles ventriculaires en temps réel pour de futures implantations sur des systèmes embarqués. Ainsi, nous avons détaillé plus notre technique choisie en présentant l'architecture matérielle des deux systèmes réalisés SYST-SP, SYST-PP ainsi que leurs étapes de fonctionnement. Ces deux modèles sont procédés à partir des classifieurs neuronaux réalisés auparavant (chapitre 4) afin d'éliminer le risque d'erreur dans le calcul.

Enfin, cette réalisation nous a permis de comparer les performances de nos modèles à d'autres méthodes qui avaient également été validées par leurs auteurs sur la base MIT-BIH en utilisant des classifieurs neuronaux matériels. Dans cette section, nous avons constaté une dissemblance dans les performances des modèles à cause de la différence des bases d'apprentissage et de test utilisées. Nos modèles ont pu obtenir des résultats acceptables, particulièrement dans la reconnaissance des extrasystoles ventriculaires.





***Conclusion Générale Et  
Perspectives.***

Avec l'évolution des densités d'intégration en microélectronique, nous sommes aujourd'hui capables de concevoir des classifieurs neuronaux sur des circuits reprogrammables du type FPGA, en vue d'une future exploitation dans un système embarqué capable de détecter les arythmies cardiaques en temps réel, permettant à l'expert d'établir un deuxième diagnostic pour confirmer la pathologie. En effet, nous avons constaté que les FPGA constituent un choix peu coûteux, accessible et flexible en les intégrant dans des dispositifs médicaux, comme ils ont également des avantages plus significatifs en ce qui concerne les implémentations neuronales.

Dans nos travaux de recherche, nous avons développé le classifieur neuronal sous l'environnement Matlab, puis nous avons réalisé la première conception matérielle du classifieur en utilisant un principe de calcul 'Semi-Parallèle'. Ce dernier nous a permis de changer l'architecture du neurone et de la couche cachée en optimisant les ressources à 22% dans le circuit avec 10 cycles d'horloge pour l'obtention d'une classe. Ensuite, nous avons procédé à la deuxième conception où nous avons optimisé le temps d'exécution à 1 cycle d'horloge seulement pour reconnaître une classe mais avec 39% de ressources utilisées dans le circuit FPGA. Alors, un compromis entre les performances relatives au temps de calcul et les besoins en ressources matérielles doit être effectué pour chaque application en fonction de ses propres contraintes.

Les contributions de cette thèse portent essentiellement sur l'emploi de ces deux principes de calcul SP (Semi-Parallèle) et PP (Parallèle-Parallèle), vu que le choix d'utiliser l'un des classifieurs réalisés dépend du vecteur de caractéristiques des données médicales à l'entrée du classifieur. En effet, dans la classification des données médicales où leurs descripteurs sont alignés séquentiellement, nous utilisons notre premier classifieur CLNH-SP, et s'ils sont placés en parallèle, nous choisissons le second classifieur CLNH-PP.

La fiabilité est une préoccupation importante pour les techniques de surveillance de la santé car la vie des gens peut dépendre de la rétroaction fournie par ces dispositifs. Afin d'améliorer la fiabilité des calculs effectués dans les classifieurs neuronaux matériels, nous avons abordé le problème du dépassement de capacité qui a été apparu en utilisant la quantification des données sur un octet. Nous avons résolu ce problème en adoptant deux protocoles de calculs expérimentaux, qui ont été utilisés pour réduire au maximum le débordement en alertant le médecin lors de son diagnostic par un système d'alarme. Après l'implémentation de nos conceptions, nous constatons que les classifieurs matériels ont apporté des améliorations considérables dans la reconnaissance des extrasystoles ventriculaires, et particulièrement en termes de temps, en éliminant tout risque de débordements dans le calcul.

Enfin, nous avons utilisé une technique en électronique câblée pour la conception de deux modèles de classification des extrasystoles ventriculaires (SYST-SP, SYST-PP) en temps réel, pour de futures implantations sur des systèmes embarqués. Cette technique consiste sur la combinaison des classifieurs neuronaux déjà réalisés afin de reconnaître l'arythmie cardiaque sans détection de dépassement de capacité. Les performances obtenues de nos deux modèles de classification sont très acceptables, où le système SYST-SP réalise un taux de

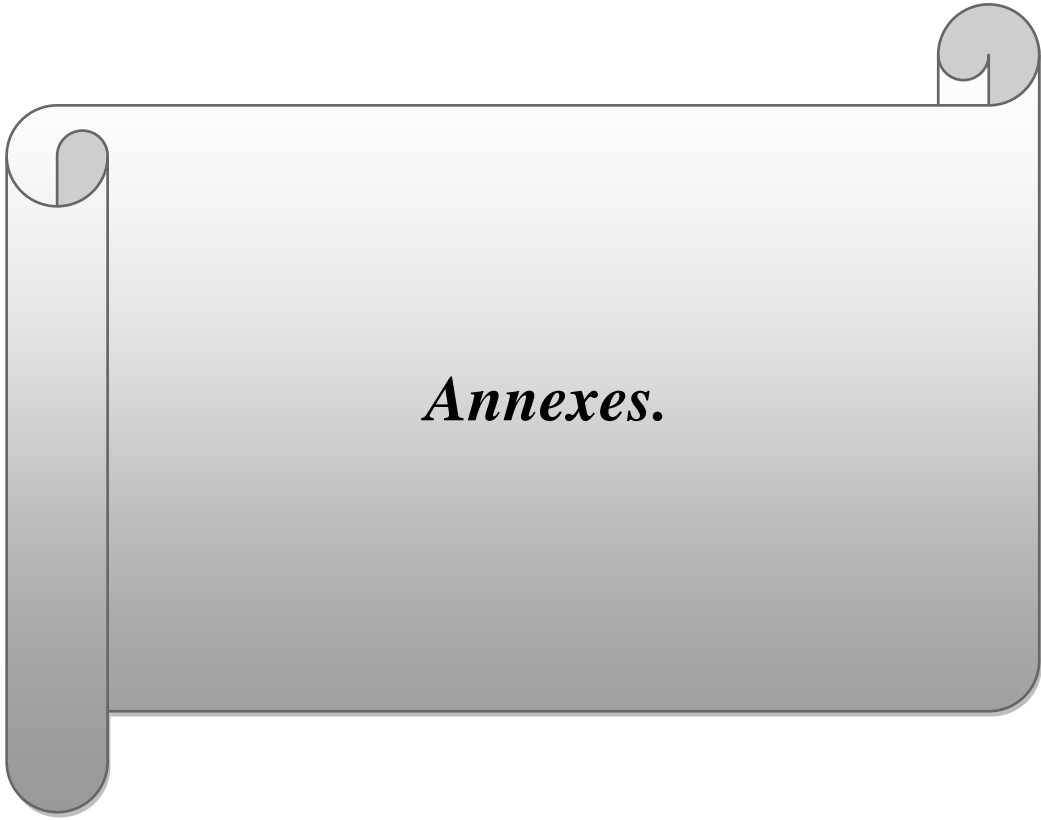
classification moyen de 92.81% (varie de 82.05% à 100%), une sensibilité moyenne de 96.94% (varie de 87.14% à 100%) et une spécificité moyenne de 94.33% (varie de 73.65% à 100%) avec une élimination totale des overflows. Tandis que le deuxième modèle SYST-PP réalise un taux de classification moyen de 92.76% (varie de 82.01% à 99.95%), une sensibilité moyenne de 96.94% (varie de 87.14% à 100%) et une spécificité moyenne de 94.95% (varie de 73.11% à 100%) sans risque de dépassement de capacité.

Les implémentations de ces modèles sur la cible FPGA peuvent être embarquées dans des dispositifs médicaux permettant la détection des arythmies cardiaques en temps réel. Les architectures étudiées s'appuient sur des contraintes d'encombrement minimaux et de souplesse maximale.

L'aspect le plus important de notre travail concerne la notion de modularité. En effet, les architectures proposées sont flexibles et facilement évolutives. Ceci a pour but de considérer d'éventuels changements dans le cahier des charges et de permettre un changement dans la reconnaissance de plusieurs pathologies. La méthode suivie nous a permis de développer une architecture générique permettant facilement de pouvoir d'adapter la structure aux conditions de classification.

Nos futurs travaux vont se focaliser sur :

- La proposition d'une classification interprétable en hybridant deux méthodes : les réseaux de neurones et la logique floue afin de réaliser une reconnaissance des données médicales, compréhensible pour l'expert en temps réel.
- L'augmentation sur la quantification des données dans l'application permet d'avoir plus d'informations et de précision, ce qui nous ramène évidemment à un taux de reconnaissance plus important.
- La grande quantité des données dans le domaine du traitement d'images médicales nécessite une résolution en temps réel, ce qui favorise le développement des implémentations matérielles à base de FPGA.
- L'utilisation de la reconfiguration partielle ou dynamique des circuits FPGA dans les applications qui nécessitent un traitement en temps réel. Par exemple, la reconfiguration dynamique (RTR: *Run Time Reconfiguration*) est utilisée souvent pour le traitement des images médicales, qui nécessite l'implémentation des algorithmes de traitement d'images de bas niveau (l'enchaînement de différents filtres) afin d'obtenir de meilleures performances de l'image.



***Annexes.***

## Annexe A

### Les statistiques des maladies

#### 1. Les maladies cardiovasculaires :

Les maladies cardiovasculaires (*Cardiovascular Diseases*) engendrent un problème majeur de la santé car elles représentent la première cause de mortalité dans le monde. En Algérie, 19,7% de décès sont dues aux maladies liées au cœur en 2008 et en 2012, le taux de mortalité a été estimé à 44,5% en arrivant à un taux de 46,2% en 2015 selon les statistiques de l'OMS (Organisation Mondiale de la Santé) [OMS\_2016].

Ces maladies constituent un ensemble de troubles affectant le cœur et les vaisseaux sanguins, ces troubles sont des anomalies qui affectent la fréquence cardiaque normale et les principales anomalies du cœur sont :

- ✚ L'extrasystole (auriculaire ou ventriculaire) avec un battement cardiaque survenant de manière précoce.
- ✚ La fibrillation (auriculaire ou ventriculaire) avec une contraction rapide et désordonnée du cœur.
- ✚ Le flutter auriculaire.

Les anomalies sont détectées par le médecin en utilisant des examens cardiologiques comme l'électrocardiogramme ECG. C'est l'examen le plus procédé car il est rapide à mettre en place et moins couteux. Le signal ECG est un tracé électrique enregistré par un appareil appelé électrocardiographe, il comporte plusieurs ondes et différents intervalles entre ces ondes (Figure A.1).

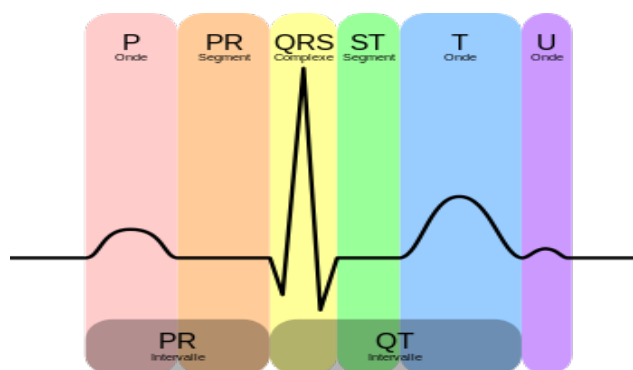


Figure A.1. Le signal ECG.

Pour des diagnostics efficaces, l'étude du signal ECG doit être effectuée pendant plusieurs heures car les symptômes des maladies peuvent se produire à tout instant, donc l'analyse et l'interprétation assistée par ordinateur des signaux ECG devient indispensable pour aider les médecins à faire le diagnostic, et pour assurer une surveillance continue des patients.

## 2. Le cancer :

Le cancer représente un phénomène tout à fait particulier dans le domaine de la santé et le nombre de patients atteints de cancer continue à augmenter de manière constante. Dans le monde, ils ont recensé 14,1 millions de nouveaux cas durant l'année 2012 avec 8,2 millions de décès déclarés dans la même année. Soit 70% de ces cas sont signalés dans les pays en développement. L'Organisation Mondiale de la Santé (OMS) prévoit que ces chiffres pourraient doubler d'ici 2030. En Algérie, les registres du cancer reconnus par les instances internationales, confirment cette tendance : actuellement, ils comptabilisent environ 45 000 nouveaux cas de cancer par an, avec 24 000 décès [OMS\_2016].

La guérison du cancer a été un objectif majeur des chercheurs médicaux pendant des décennies, et en sachant que le développement de nouveaux traitements prend du temps et de l'argent, les chercheurs ont développé différents techniques pour la détection précoce de la maladie. Dans notre étude, nous présentons seulement deux types de cancer : le cancer du sein (*Breast Cancer*) et le cancer du cerveau (*Brain Cancer*).

### 2.1. Le cancer du sein :

Le cancer du sein est le cancer le plus souvent diagnostiqué chez les femmes, il présente la principale cause de décès. Dans le monde et selon l'OMS, 1.6 millions nouveaux cas a été estimé en 2012, soit 25% de l'ensemble des cancers de la femme. En Algérie, 11000 nouveaux cas détecté par année, il représente plus de 40% de l'ensemble des cancers de la femme [ZITOUNI\_2014].

L'augmentation rapide et importante du nombre de cancers du sein est un phénomène inquiétant et qui mérite un développement particulier. Le diagnostic précoce de cette maladie à l'heure actuelle, est la seule arme pour améliorer l'état des malades, il permet de réduire le taux de mortalité par ce cancer, et d'assurer une meilleure survie des patientes voire une guérison totale. Il est admis qu'un dépistage de qualité du cancer du sein entraîne une baisse de la mortalité de 15 à 30%. Le dépistage du cancer du sein repose actuellement sur la mammographie qui est l'examen de référence, couplée à l'échographie voire à l'IRM dynamique dans certaines indications bien précises.

La mammographie est le meilleur outil de dépistage qui utilise des rayons X de faible dose pour créer une image du sein afin de détecter le cancer. L'ACS (*American Cancer Society*) recommande que toutes les femmes âgées de plus de 40 ans subissent une mammographie de dépistage une fois par an. Dans ces images, les tissus denses du sein peuvent paraître blancs ou gris clair, ce qui peut rendre les mammographies plus difficiles à interpréter chez les femmes plus jeunes car elles ont des seins plus denses. Donc en fonction de ces derniers, les radiologues peuvent manquer jusqu'à 30% des cancers du sein [KOLB\_2002].

Deux indicateurs puissants de cancer du sein qui sont couramment utilisés dans l'évaluation des mammographies : les masses et les micro-calcifications (petites accumulations de calcium qui peuvent être présentes dans le tissu mammaire). La détection de masse est un problème plus difficile que la recherche de micro-calcifications, car les masses peuvent

apparaître dans une mammographie avec différentes tailles et formes dont elles se présentent en mauvais contraste d'image [CHENG\_2003], d'où les radiologistes ont du mal à interpréter ces mammographies même les plus qualifiés. Alors la détection et le diagnostic automatisé seront primordiaux dans le traitement des images numériques.

## **2.2. Le cancer du cerveau :**

Les tumeurs cérébrales sont des cancers rares. En 2010, le nombre estimé était de 4570 nouveaux cas de tumeurs cérébrales, soit 1.3% de l'ensemble des cancers et le nombre de décès estimé était de 2 940, soit 2% de l'ensemble des décès par cancer [INCA\_ 2010]. Bien qu'elles soient plus fréquentes chez les personnes âgées, les tumeurs cérébrales touchent les adultes à tout âge ainsi que les enfants. Chez ces derniers, c'est le cancer le plus fréquent après la leucémie (23% des cancers pédiatriques) [SOMMELET\_2009].

Une tumeur cérébrale est toute masse qui résulte d'une croissance anormale et incontrôlée des cellules dans le cerveau. Son niveau de menace dépend d'une combinaison de facteurs tels que le type de tumeur, son emplacement, sa taille et son état de développement. Les tumeurs du cerveau peuvent être cancéreuses (malignes) ou non cancéreuses (bénignes), ces derniers sont de faible grade, ils poussent lentement en repoussant le tissu normal, ils sont homogènes, bien définis et ils sont connus sous le nom de tumeurs non métastatiques, car ils ne forment pas de tumeurs secondaires. Alors que les tumeurs cérébrales malignes sont des tumeurs cancéreuses, qui poussent rapidement et envahissent le tissu normal, ils peuvent être comptés parmi les maladies les plus mortelles.

L'imagerie cérébrale, en particulier l'IRM (Imagerie par Résonance Magnétique) cérébrale a été un progrès majeur dans le diagnostic et la prise en charge des tumeurs cérébrales, il a une sensibilité de contraste qui permet de distinguer les différentes catégories de tissus mous et de préciser l'anatomie et la localisation des lésions, cette technologie fournit une imagerie de résolution identique et dans les trois plans de l'espace. Les IRM cérébrales correspondent à l'utilisation d'un champ magnétique, ils sont principalement de 1.5 Tesla, mais la plupart des établissements sont équipés d'IRM à 3 Tesla (Figure A.2) qui permettent une grande précision pour visualiser les structures cérébrales. Mais malheureusement, l'IRM est un examen long car il comprend plusieurs séquences et chaque séquence dure de deux à cinq minutes, et durant l'acquisition des images, n'importe quel mouvement du patient rend les images ininterprétables [INFO\_2015].



**Figure A.2. Système d'IRM à 3 Tesla.**

### **3. Le diabète :**

Le diabète est une maladie chronique qui ne cesse de progresser à l'échelle mondiale et selon l'OMS, si cette progression continue le diabète serait la 7<sup>ème</sup> cause de décès dans le monde d'ici 2030. En 2012, ils ont estimé 1.5 millions de décès étaient directement dus au diabète et que 2.2 millions de décès supplémentaires devaient être attribués à l'hyperglycémie. Le nombre de personnes atteintes de diabète est passé de 108 millions en 1980 à 422 millions en 2014 [OMS\_2016].

En Algérie, le diabète vient en deuxième position au classement des maladies chroniques derrière l'hypertension, selon la 3<sup>ème</sup> étude nationale des indications multiples. Le nombre de personnes atteintes de diabète est en progression, passant de 0.3% chez les sujets âgés de moins de 35 ans à 4.1% chez les 35-59 ans, et 12.5% chez les plus de 60 ans, selon cette étude menée par le ministère de la Santé, de la Population et de la Réforme hospitalière, en collaboration avec l'Office nationale des statistiques et des représentations des Nations unies à Alger.

Le diabète est une maladie chronique qui apparaît lorsque le pancréas ne produit pas suffisamment d'insuline ou que l'organisme n'utilise pas correctement l'insuline qu'il produit. L'insuline est une hormone qui régule la concentration de sucre dans le sang. L'hyperglycémie, ou concentration sanguine élevée de sucre, est un effet fréquent du diabète non contrôlé qui conduit avec le temps à des atteintes graves de nombreux systèmes organiques et plus particulièrement des nerfs et des vaisseaux sanguins [WHO\_1999].

### **4. Les troubles neurologiques :**

Les troubles neurologiques sont des maladies du système nerveux central ou périphérique. En d'autres termes ils touchent le cerveau, la moelle épinière, les nerfs crâniens, les nerfs périphériques, les racines nerveuses, le système nerveux végétatif, la jonction neuromusculaire et les muscles.

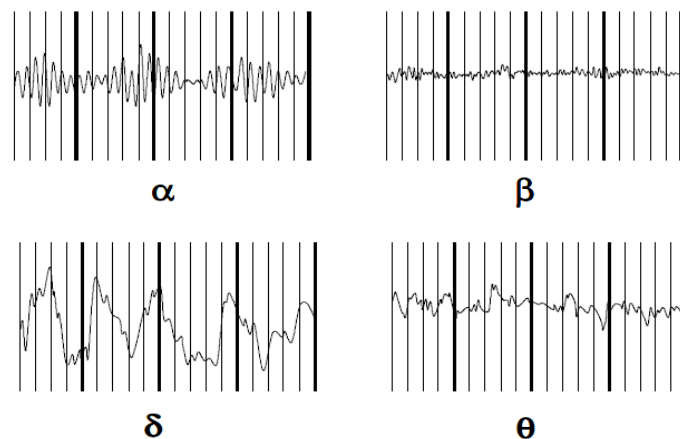


Ces troubles comprennent l'épilepsie, la maladie de Parkinson, la maladie d'Alzheimer, et d'autres démences. Le signal d'électroencéphalogramme (EEG) est largement utilisé cliniquement pour étudier ces troubles cérébraux, c'est un examen qui permet d'enregistrer l'activité cérébrale, au moyen d'électrodes qui sont placées sur la surface du cuir chevelu et des différences de potentiel électrique produites au niveau du cerveau [OMS\_2016].

L'électro-encéphalographie est l'enregistrement graphique, au moyen d'électrodes qui sont placées sur la surface du cuir chevelu, des différences de potentiel électrique, produites au niveau du cerveau. Cet examen est indolore et sans danger. Son utilisation a permis de mieux comprendre le fonctionnement cérébral, il trouve son exploitation principale dans la détection et le suivi de certaines maladies du système nerveux comme l'épilepsie, de troubles de la compréhension ou encore des conséquences d'hémorragies cérébrales. Aujourd'hui, l'encéphalographie permet même d'étudier le sommeil d'un individu.

L'EEG enregistre des ondes cérébrales ou "activités EEG" (Figure A.3) qui sont caractérisées par leur fréquence, leur amplitude, leur stabilité, leur morphologie, leur topographie, leur réactivité [ANTONIA\_2006]. Elles sont classées en fonction de leur bande de fréquences en :

- ✚ Ondes delta (fréquence inférieure à 3,5 Hz).
- ✚ Ondes thêta (fréquence comprise entre 4 et 7,5 Hz).
- ✚ Ondes alpha (fréquence comprise entre 8 et 13 Hz).
- ✚ Ondes bêta (fréquence supérieure à 13 Hz).



**Figure A.3. Principaux rythmes de l'EEG.**

#### **4.1. Maladies de Parkinson :**

La maladie de Parkinson est un trouble neurologique progressif caractérisé par un tremblement, une rigidité et une lenteur des mouvements. Il est associé à une perte neuronale progressive de la substance noire et d'autres structures cérébrales. Cette situation conduit à la nécessité d'élaborer un système d'aide à la décision pour une détection précoce de cette maladie [ENCYCLO\_2016].

#### **4.2. L'épilepsie :**

L'épilepsie est l'une des affections neurologiques les plus fréquentes, dont on peut souffrir à n'importe quel âge. Dans le monde, environ 50 millions de personnes sont atteintes d'épilepsie, et près de 80% des personnes qui vivent dans les pays en développement [OMS\_2016].

L'épilepsie se caractérise par des décharges d'influx nerveux anormaux dans le cerveau. Ces décharges surviennent de façon soudaine. Habituellement, elles sont de courte durée. Elles peuvent avoir lieu soit dans une zone précise du cerveau, soit dans son ensemble. Ces influx nerveux anormaux peuvent se mesurer durant un électro-encéphalogramme (EEG) [EPILEPT\_2001].

## Annexe B

### Programmes principaux réalisés sous Matlab.

#### Pour le test du classifieur neuronal :

% En utilisant tic au début et toc à la fin du programme, je calcule le temps de classification.  
tic

% Ici je charge les données propres au malade.

load pvcInputs

load pvcTargets

% Ici je charge les paramètres de mon réseau.

load net\_essai10;

% Test du Réseau

%=====

% ICI je prends les paramètres du malade pour la classification

inputs = pvcInputs;

targets = pvcTargets;

%=====

outputs = net(inputs);

errors = gsubtract(targets,outputs);

performance = perform(net,targets,outputs);

%===== CALCUL =====

outputs\_r=round(outputs);

errors2 = gsubtract(targets,outputs\_r);

% Calcul du nombre de cas reconnus.

Nbre=length(find(targets==outputs\_r));

disp('nombre de cas reconnus est : ');disp(nbre) ;

% Calcul du taux de classification.

Taux=(length(find(targets==outputs\_r))/length(targets))\*100;

disp('taux de reconnaissance est : ');disp(taux) ;

N=length(targets);

% Calcul des prédicts positifs et négatifs qui le sont vraiment.

Tp1=0;

tn1=0;

for i=1:N

if (targets(i)==-1)&(outputs\_r(i)==-1)

tp1=tp1+1;

end;

if (targets(i)==1)&(outputs\_r(i)==1)

tn1=tn1+1;

```
end;
end;

% Calcul des prédicts positifs qui sont en fait négatifs et
% calcul des prédicts négatifs qui sont en fait positifs.
Fp1=0;
fn1=0;
for i=1:N
    if (targets(i)==1)&(outputs_r(i)==-1)
        fp1=fp1+1;
    end;
    if (targets(i)==-1)&(outputs_r(i)==1)
        fn1=fn1+1;
    end;
end;
format short;
disp('Le nombre de cas TP1 est :') ;disp(tp1) ;
disp('Le nombre de cas TN1 est :') ;disp(tn1) ;
disp('Le nombre de cas FP1 est :') ;disp(fp1) ;
disp('Le nombre de cas FN1 est :') ;disp(fn1) ;
toc.
```



***Références Bibliographiques.***

- [ABIBULLAEV\_2011] Berdakh Abibullaev, Hee Don Seo. (August 2011). ‘**A New QRS Detection Method Using Wavelets and Artificial Neural Networks**’. *Journal of Medical Systems*, Volume 35, Issue 4, pp 683–691.
- [AKANKSHA\_2016] Akanksha Nilosey. (October 2016). ‘**FPGA Based Diabetic Patient Health Monitoring Using Fuzzy Neural Network**’. *International Journal of Science and Research (IJSR)* ISSN (Online): 2319-7064. Volume 5 Issue 10.
- [ALTERA\_2002] [https://www.altera.com/literature/ds/archives/ds\\_ap2](https://www.altera.com/literature/ds/archives/ds_ap2).
- [AMSAVENI\_2014] V. Amsaveni, N. Albert Singh. (2014). ‘**Detection of brain tumor using neural network**’. *Computing, Communications and Networking Technologies (ICCCNT), Fourth International Conference on*. Date Added to IEEE Xplore: 30 January 2014.
- [ANAND\_1992] Anand, Mehrotra, Mohan & Ranka. (Avril 1992). ‘**Intelligent Control Using Neural Networks**’. *IEEE Control System mag*, pp 11-18.
- [ANIKET\_2013] Aniket A. Kathalkar, R. S. Kawitkar, Amruta Chopade. (March 2013). ‘**Artificial Neural Network based Brain Cancer Analysis and Classification**’. *International Journal of Computer Applications* (0975 – 8887) Volume 66– No.10.
- [ANTONIA\_2006] Antonia Pérez-Martin, Iris Schuster, Michel Dauzat. (Novembre 2006). ‘**Electroencéphalogramme**’. *PCEM 2 - Module de Base 6. Travaux Pratiques de Physiologie*. Faculté de Médecine Montpellier-Nîmes. En partenariat avec les Editions SAURAMPS Médical – Montpellier.
- [AQHSA\_2014] Aqhsa Q. Syed, K. Narayanan. (September 2014). ‘**Detection of Tumor in MRI Images Using Artificial Neural Networks**’. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 9. ISSN (Print): 2320 – 3765 ISSN (Online): 2278 – 8875.
- [ARLOT\_04/2014] Pierrick Arlot. (23/04/2014). ‘**Altera Câble dans le Silicium de ses FPGA des Blocs de Calcul en Virgule Flottante**’. L’embarqué, Logiciel & Systèmes.
- [ARLOT\_12/2014] Pierrick Arlot. (19/12/2014). ‘**Xilinx et Altera s’asticotent sur le Marché des FPGA Gravés en 20nm**’. L’embarqué, Logiciel & Systèmes.
- [ARMATO\_2009] A. Armato, E. Nardini, A. Lanatà, G. Valenza, C. Mancuso, E.P. Scilingo, D. De Rossi. (2009). ‘**An FPGA based arrhythmia recognition system for wearable applications**’. *Ninth International Conference on Intelligent Systems Design and Applications*.

- [ASYIQIN\_2010] N. A. A. Hamzah and R. Besar. (2010). **‘Comparison of different mother wavelets in pvc detection using pnn’**. In *Information Sciences Signal Processing and their Applications (ISSPA)*, 2010 10th International Conference on. IEEE, pp. 496–499.
- [BABA\_06/2014] Baba Hamed Amel, Bechar Hassane. (Juin 2014). **‘Description Matérielle Des Circuits Numériques En Langage VHDL’**. *Manuel des Travaux Pratiques*. Editions internes de la faculté de l’université de Tlemcen.
- [BABA\_10/2014] Baba Hamed Amel, Bechar Hassane, Chikh Mohammed Amine. (15-16 octobre 2014). **‘Conception Matérielle D’un Classifieur Neuronal En Technologie FPGA En Vue D’une Classification Médicale’**. *Biomedical Engineering International Conference (BIOMEIC’14)*. Tlemcen.
- [BABA\_2016] A. Baba Hamed, H. Bechar, M.A. Chikh. (March 2016). **‘Implementation of a neural classifier on a FPGA-based reconfigurable of cardiac arrhythmias’**. *EEA: Electrotehnica, Electronica, Automatica*, Vol.64, Nr.1, PP.177-186. ISSN: 1582-5175.
- [BELAROUCI\_2016] Sara Belarouci. (Juin 2016). **‘Traitement et classification intelligente des données médicales non équilibrées’**. THÈSE de doctorat.
- [BENCHAIB\_2014] Y. Benchaib , M.A. Chikh. (2014). **‘Cardiac arrhythmias classification using artificial metaplasticity algorithm’**. *Int. J. of Biomedical Engineering and Technology* 2014 - Vol. 14, No.3 pp. 209 - 224.
- [BENEDIC\_2007] Yohann BÉNÉDIC. (2007). **‘Approche Analytique Pour L’optimisation De Réseaux De Neurones Artificiels’**. *THÈSE de Doctorat*. Sciences de l’ingénieur. Université de Haute Alsace – Mulhouse.
- [BEUCHAT\_2001] Jean-Luc Beuchat. (2001). **‘Etude Et Conception D’opérateurs Arithmétiques Optimisés Pour Circuits Programmables’**. THÈSE No 2426.
- [BLOTIN\_2005] T. BLOTIN. (2005). **‘Le Langage De Description VHDL’**. Lycée Paul-Eluard, 93206 Saint-Denis.
- [BOUSSAA\_2015] Boussaa Mohamed , Atouf Issam, Atibi Mohamed, Bennis Abdellatif. (2015). **‘ECG Image Classification in Real time based on the Haar-like Features and Artificial Neural Networks’**. *Procedia Computer Science, International Conference on Advanced Wireless Information and Communication Technologies (AWICT 2015)*. Volume 73, Pages 32-39.

- [CAN\_2013] Mehmet Can. (2013). **‘Neural Networks to Diagnose the Parkinson’s Disease’**. *Southeast Europe Journal Of Soft Computing*. Vol: 2, No 1. pp 68-75.
- [CARRE\_1984] A. Carré. (1984). **‘Révision Accélérée en Cardiologie’**. Maloine S.A éditeur Paris.
- [CHANDEL\_2013] Deepali Chandel, Gagan Kumawat, Pranay Lahoty, Vidhi Vart Chandrodya, Shailendra Sharma. (March 2013). **‘Booth Multiplier: Ease of Multiplication’**. *International Journal of Emerging Technology and Advanced Engineering*, ISSN 2250-2459, Volume 3, Issue 3.
- [CHENG\_2003] H.D. Cheng, X. Cai, X. Chen, L. Hu, X. Lou. (2003). **‘Computer-aided detection and classification of microcalcifications in mammograms: a survey’**. *Pattern Recogn*, 36 (12), pp. 2967–2991.
- [CHIKH\_2004] Chikh Mohammed Amine. (Juillet 2004). **‘Reconnaissance Des Battements Ventriculaires Prématursés Par Les Réseaux De Neurones Et La Logique Floue’**. THÈSE de Doctorat.
- [CLERET\_2001] Cléret Mireille, Le Beux Pierre, Le Duff Franck. (2/2001). **‘Les systèmes d’aide à la décision médicale’**. *Les Cahiers du numérique*, Vol. 2, p. 125-154.
- [DARCHE\_2004] Phillip DARCHE. (2004). **‘Architecture Des Ordinateurs, Logique booléenne : implémentations et technologies’**. Vuibert, Paris, ISBN 2-7117-4821-9.
- [DAS\_2010] Resul Das. (March 2010). **‘A comparison of multiple classification methods for diagnosis of Parkinson disease’**. *Expert Systems with Applications*. Volume 37, Issue 2, Pages 1568–1572. Elsevier.
- [DETREY\_2007] Jérémie DETREY. (15 Janvier 2007). **‘Arithmétiques réelles sur FPGA. Virgule fixe, virgule flottante et système logarithmique’**. Thèse De Doctorat, L’École Normale Supérieure De Lyon.
- [DEVICE\_2012] DeviceMed France. (2012). **‘Système d’Imagerie Médicale par OCT 3D en Temps Réel’**. [www.devicemed.fr](http://www.devicemed.fr).
- [DHANDE\_2015] J. D. Dhande, Dr. S. M. Gulhane. (May 2015). **‘Neural Networks Design for Classification of Epilepsy EEG Signals’**. *International Journal of Science and Research (IJSR)* ISSN (Online): 2319-7064. Volume 4 Issue 5.
- [DHEEBA\_2014] J. Dheebea, N. Albert Singh, S. Tamil Selvi imagerie. (June 2014). **‘Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach’**. *Journal of Biomedical Informatics*. Volume 49, Pages 45–52.



- [DILLIEN\_2017] Paul Dillien. (2017). ‘**And the Winner of Best FPGA of 2016 is...**’. *Principal, High Tech Marketing*. EE | Times Connecting the Global Electronics Community.
- [DIPALI\_2010] Dipali M. Joshi , N. K. Rana, V. M. Misra. (2010). ‘**Classification of Brain Cancer using Artificial Neural Network**’. *Published in: Electronic Computer Technology (ICECT), 2010 International Conference on*. Date Added to IEEE Xplore: 07 June 2010.
- [DOCUSSE\_2013] Tiago A. Dócusse, Alexandre C.R. da Silva, Aledir S. Pereira, Norian Marranghello. (2013). ‘**Development of a CAD System for Automatic Classification of Microcalcifications based on FPGA**’. *AASRI Procedia. AASRI Conference on Intelligent Systems and Control*. Volume 4, Pages 90-95.
- [DOGANTEKIN\_2010] Dogantekin E, Akif D, Derya A, Levent A. (July 2010). ‘**An intelligent diagnosis system for diabetes on linear discriminant analysis and adaptive network based fuzzy inference system: LDA–ANFIS**’. *Digital Signal Processing*. Volume 20, Issue 4, Pages 1248-1255.
- [DOUILLARD\_2005] Catherine Douillard, Gérald Ouvradou, Michel Jezequel. (2005). ‘**ELP304/203 : Electronique Numérique, Logique Séquentielle Techniques D’intégration**’. *Telecom Bretagne*.
- [DUTRIEUX\_1997] Laurent Dutrieux, Didier Demigny. (1997). ‘**Logique Programmable, Architecture des FPGA et CPLD**’. *Edition Eryrolles, ISBN 2-212-09581-3*.
- [EBRAHIMZADEH\_2010] A. Ebrahimzadeh and A. Khazae. (2010). ‘**Detection of premature ventricular contractions using mlp neural networks: a comparative study**’. *Measurement*, vol. 43, no. 1, pp. 103–112.
- [EGILA\_2016] Mohamed G. Egila, Magdy A. El-Moursy, Adel E. El-Hennawy, Hamed A. El-Simary, Amal Zaki. (2 April 2016). ‘**FPGA-based electrocardiography (ECG) signal analysis system using least-square linear phase finite impulse response (FIR) filter**’. *Journal of Electrical Systems and Information Technology* 3, 513-526.
- [EL MOUKHLIS-2014] Sami El Moukhlis, Abdessamad Elrharras And Abdellatif Hamdoun. (March 2014). ‘**FPGA Implementation of Artificial Neural Networks**’. *IJCSI International Journal of Computer Science*, Issues, Vol. 11, Issue 2, No 1.
- [ENCYCLO\_2016] [http://www.doctissimo.fr/html/sante/encyclopedie/sa\\_1085\\_parkinson.htm](http://www.doctissimo.fr/html/sante/encyclopedie/sa_1085_parkinson.htm).
- [EPILEPT\_2001] <http://epileptologie-bonn.de/cms>.

- [ETATS\_2015] ETATS-UNIS, Rachat, Semiconducteur. (2015). **‘Intel négocie le rachat d’Altera pour plus de 10 milliards de dollars’**. L’électronique au quotidien.
- [GAUTHIER\_2013] François Gauthier. (2013). **‘Le Marché des FPGA pourrait croître de 8.5% par an jusqu’en 2019 pour atteindre les 9 Md\$’**. L’embarqué, Logiciel & Systèmes.
- [GHORBANIAN\_2011] P. Ghorbanian, A. Jalali, A. Ghaffari, and C. Nataraj. (2011). **‘An improved procedure for detection of heart arrhythmias with novel preprocessing Techniques’**. *Expert systems*, vol. 29, no. 5, pp. 478–491.
- [GHOZZI\_2003] Fahmi GHOZZI. (2003). **‘Optimisation D’Une Bibliothèque De Modules Matériels De Traitement D’Images. Conception Et Test VHDL, Implémentation Sous Forme FPGA’**. Thèse De Doctorat, L’université Bordeaux I.
- [GONZALEZ\_2013] Carlos M. Travieso-González, Santiago T. Pérez-suárez, Jesús B. Alonso. (2013). **‘Using Fixed Point Arithmetic for Cardiac Pathologies Detection Based on Electrocardiogram’**. *Computer Aided Systems Theory-EUROCAST* . Volume 8112 of the series Lecture Notes in Computer Science pp 242-249. Series ISSN: 0302-9743.
- [GOTHWAL\_2011] H. Gothwal, S. Kedawat, and R. Kumar (2011). **‘Cardiac Arrhythmias Detection in an ECG Beat Signal Using Fast Fourier Transform and Artificial Neural Network’**. *Journal of Biomedical Science and Engineering*, Vol.4, pp.289-296.
- [GOURI\_2015] Gouri B. Deshpande and Dr.K. Ramesha. (May 2015). **‘MRI Brain Image Enhancement Using XILINX System Generator and DWT’**. *Bonfring International Journal of Advances in Image Processing*, Vol. 5, No. 2.
- [GUEX\_1998] C. GUEX & E. MESSERLI. (Janvier 1998). **‘Circuits Programmables Et Langages De Conception, Une Evolution En Parallèle’**. *Revue Visions, Revue scientifique de l’EIVD*, page 49.
- [HAITHAM\_2010] Haitham Kareem Ali, Esraa Zeki Mohammed. (August 2010). **‘Design Artificial Neural Network Using FPGA’**. *IJCSNS International Journal of Computer Science and Network Security*, VOL.10 No.8,
- [HARIKUMAR\_2013] Dr. R Harikumar, M. Balasubramani, S. Saravanan. (March 2013). **‘FPGA Implementation of Wavelet Neural Network for Epilepsy Detection’**. *International Journal of Engineering and Innovative Technology (IJEIT)*. Volume 2, Issue 9. ISSN: 2277-3754.
- [HASSANIEN\_2014] Aboul Ella Hassanién, Hossam M. Moftah, Ahmad Taher Azar, Mahmoud Shoman Imagerie. (January 2014). **‘MRI breast**

- cancer diagnosis hybrid approach using adaptive ant-based segmentation and multilayer perceptron neural networks classifier**'. *Applied Soft Computing. Special issue on hybrid intelligent methods for health technologies*. Volume 14, Part A, Pages 62–71.
- [HEMANTH\_2011] Hemanth D.Jude, Vijila C.Kezi Selva, Anitha J. (2011). **'Application of Neuro-Fuzzy Model for MR Brain Tumor Image Classification'**. *Biomedical Soft Computing and Human Sciences*, Vol.16, N°.1, pp.95-102.
- [HENDEL\_2010] M. Hendel, A. Benyettou, F. Hendel, H. Khelil. (2010). **'Automatic heartbeats classification based on discrete wavelet transform and combination of probabilistic MLP neuronal networks'**. *The international Journal of Applied Engineering research (IJAER)*, Vol 1, Number 11, PP. 1925-1936.
- [INCA\_2010] (juin 2010). **'Les tumeurs du cerveau'**. *Collection Guides patients Cancer info*. INCa.
- [INFO\_2015] [http://www.info-radiologie.ch/irm\\_cerebrale.php](http://www.info-radiologie.ch/irm_cerebrale.php).
- [JADHAV\_2012] Jadhav, S.M, Nalbalwar, S.L. and Ghatol, A.A. (2012). **'Artificial neural network models based cardiac arrhythmia disease diagnosis from ECG signal data'**. *International Journal of Computer Applications*, Vol. 44, No. 15, pp.8–13.
- [JANG\_1991] Jang, Jyh-Shing R (1991). **'Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm'**. *Proceedings of the 9th National Conference on Artificial Intelligence*, Anaheim, CA, USA, July 14–19. 2. pp. 762–767.
- [JATMIKO\_2011] W. Jatmiko, P. Mursanto, A. Febrian, M. Fajar, W.T. Anggoro, R.S. Rambe, M.I. Tawakal, F.F. Jovan, S. Eka, (November 2011). **'Arrhythmia classification from wavelet feature on FGPA'**. In: *IEEE International Symposium on Micro-NanoMechatronics and Human Science*, Nagoya, Japan, pp. 349–354.
- [JEMBULA\_2013] K.K. Jembula, G. Srinivasulu, K.S. Prasad. (May 2013). **'Design Of Electrocardiogram (ECG Or EKG) System On FPGA'**. *Research Inveny: International Journal Of Engineering And Science*, Vol.3, Issue 2, PP 21-27.
- [JEWAJINDA\_2010] Yutana Jewajinda, Prabhas Chongstitvatana. (2010). **'FPGA-based Online-learning using Parallel Genetic Algorithm and Neural Network for ECG Signal Classification'**. *Published in: Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*. Date Added to IEEE Xplore: 24 June 2010.

- [JEWAJINDA\_2013] Yutana Jewajinda, Prabhas Chongstitvatana. (June 2013). ‘**A parallel genetic algorithm for adaptive hardware and its application to ECG signal classification**’. *Neural Computing and Applications*. Volume 22, Issue 7, pp 1609–1626.
- [KARABERNOU\_2009] Si Mahmoud KARABERNOU. (Septembre 2009). ‘**Méthodologie De Conception Et Langages De Description De Matériel**’. Ecole Nationale Supérieure De L’Electronique Et De Ses Applications (ENSEA).
- [KARABERNOU\_2016] Si Mahmoud KARABERNOU. (Septembre 2016). ‘**Introduction, Systèmes sur Puce**’. Ecole Nationale Supérieure De L’Electronique Et De Ses Applications (ENSEA).
- [KOLB\_2002] T.M. Kolb, J. Lichy, J.H. Newhouse. (2002). ‘**Comparison of the performance of screening mammography, physical examination, and breast US and evaluation of factors that influence them: an analysis of 27,825 patient evaluations Radiology**’. 225 (1), pp. 165–175.
- [KUEH\_2016] Si Mon Kueh, Tom J. Kazmierski. (2016). ‘**Massively-Parallel Bit-Serial Neural Networks for Fast Epilepsy Diagnosis: A Feasibility Study**’. *World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering*. Vol: 10, No: 1.
- [MAASS\_1997] Maass, Wolfgang. (1997). ‘**Networks of spiking neurons: The third generation of neural network models**’. *Neural Networks*. 10 (9): 1659–1671. ISSN 0893-6080
- [MARCANO\_2011] A. Marcano-Cedeño, J. Quintanilla-Domínguez, D. Andina. (August 2011). ‘**WBCD breast cancer database classification applying artificial metaplasticity neural network**’. *Expert Systems with Applications*. Volume 38, Issue 8, Pages 9573–9579.
- [MARK\_1997] V. G. Mark, H. Jansen. (1997). ‘**Using artificial neural network for classifying ICU patient states**’. *IEEE EMB Mag.* 41–47.
- [MENAKA\_2014] Dr. R. Menaka, S. Rohini. (2014). ‘**Efficient Detection of Ischemic Stroke from MRI Images Using Wavelet Transform**’. *International Journal of Computer Science and Information Technology Research*. Vol. 2, Issue 3, pp: (446-454), Month: July – September. ISSN 2348-1196.
- [MESSAOUD\_2005] A. Messaoud, M. Ben Messaoud, A. Kachouri and F. Sellami. (2005). ‘**Classification des arythmies cardiaques par logique floue à partir de signaux ECG**’. SETIT, *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, March 27-31, TUNISIA.

- [MIT\_2000] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. (June 13.2000). **'PhysioBank, PhysioToolkit and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals'**. *Circulation* 101(23): e215-e220. <http://www.physionet.org/physiobank/database>.
- [NITHYA\_2011] R. Nithya, B. Santhi. (August 2011). **'Classification of Normal and Abnormal Patterns in Digital Mammograms for Diagnosis of Breast Cancer'**. *International Journal of Computer Applications* (0975 – 8887) Volume 28– No.6.
- [NNT\_2016] Neural Network Toolbox - MATLAB - MathWorks (2016). [www.mathworks.com/products/neural-network](http://www.mathworks.com/products/neural-network).
- [OLANREWAJU\_2015] Rashidah. Funke Olanrewaju, Nur Syarafina Sahari, Aibinu A. Musa, Nashrul Hakiem. (2015). **'Application of neural networks in early detection and diagnosis of Parkinson's disease'**. *Published in: Cyber and IT Service Management (CITSM), 2014 International Conference on*. Date Added to IEEE Xplore: 16 February 2015.
- [OMERHODZIC\_2013] Ibrahim Omerhodzic, Samir Avdakovic, Amir Nuhanovic, Kemal Dizdarevic. (30 Jul 2013). **'Energy Distribution of EEG Signals: EEG Signal Wavelet-Neural Network Classifier'**. *Neural and Evolutionary Computing, World Academy of Science, Engineering and Technology*, 61, 1190-1195.
- [OMS\_2016] **'Global Atlas on Cardiovascular Disease Prevention and Control'**. *World Health Organization*, Geneva.
- [ORHAN\_2016] Er Orhan; Cetin Onursal; Bascil M. Serdar; Temurtas Feyzullah. (February 2016). **'A Comparative Study on Parkinson's Disease Diagnosis Using Neural Networks and Artificial Immune System'**. *Journal of Medical Imaging and Health Informatics*, Volume 6, Number 1, pp. 264-268(5). Publisher: American Scientific Publishers.
- [OTHMAN\_2011] Mohd Fauzi Othman, Mohd Ariffanan Mohd Basri. (2011). **'Probabilistic Neural Network for Brain Tumor Classification'**. *Published in: Intelligent Systems, Modelling and Simulation (ISMS), 2011 Second International Conference on*. Date Added to IEEE Xplore: 14 March 2011. ISBN: 978-1-4244-9809-3.
- [OUELLI\_2015] Abdelhaq Ouelli, Benachir Elhadadi, Hicham Aissaoui, Belaid Bouikhalene. (2015). **'Epilepsy Seizure Detection Using Autoregressive Modelling and Multiple Layer Perceptron Neural Network'**. *American Journal of Computer Science and Engineering*. Vol. 2, No. 4, pp. 26-31.

- [OURDIGHI\_2016] Ourdighi Asmaa, Benyettou Abdelkader. (May2016). ‘**An Efficient Spiking Neural Network Approach based on Spike Response Model for Breast Cancer Diagnostic**’. *International Arab Journal of Information Technology (IAJIT)*. Vol. 13 Issue 3, p284-290. 7p.
- [ÖZBAY\_2011] Yüksel Özbay, Rahime Ceylan, Bekir Karlik. (January 2011). ‘**Integration of type-2 fuzzy clustering and wavelet transform in a neural network based ECG classifier**’. *Expert Systems with Applications*, Volume 38, Issue 1, Pages 1004–1010.
- [OZDEMIR\_2011] Ahmet Turan OZDEMIR, Kenan DANISMAN. (2011). ‘**Fully parallel ANN-based arrhythmia classifier on a single-chip FPGA: FPAAC**’. *Turk J Elec Eng & Comp Sci*, Vol.19, No.4.
- [PAN\_1985] Pan J, Tompkins WJ. (1985). ‘**A Real-Time QRS Detection Algorithm**’. *IEEE T Biomed Eng.* 32(3):230-6.
- [PATRA\_2010] Dipti Patra, Manab Kumar Das, Smita Pradhan. (3 February 2010). ‘**Integration of FCM, PCA and Neural Networks for Classification of ECG Arrhythmias**’. *IAENG International Journal of Computer Science*, 36:3, IJCS\_36\_3\_05.
- [PINEDA\_1987] F.J. Pineda. (1987). ‘**Generalization Of Back-Propagation To Recurent Neural Networks**’. *Physical Review letters*, 59(19), pp 2229-2232.
- [PINJARE\_2012] S L Pinjare and Arun Kumar M. (August 2012). ‘**Implementation of Neural Network Back Propagation Training Algorithm on FPGA**’. *International Journal of Computer Applications* 52(6):1-7. Published by Foundation of Computer Science, New York, USA.
- [PISTES\_2009] Pistes et Pastilles. (Janvier 2009). ‘**Actel Répond à la Tendance vers la Miniaturisation et la Portabilité de l’Équipement Médical**’. *ELECTRONIQUE Mag.com*.
- [POUSSIÉ\_2002] Sylvain POUSSIÉ & Hassan RABAH & Serge WEBER. (2002). ‘**SOPC-based Embedded Smart Strain Gage Sensor**’. (Eds.): FPL 2002, LNCS 2438, pp. 1131–1134. c\_Springer-Verlag Berlin Heidelberg.
- [PRADHAN\_2011] Manaswini Pradhan, Dr. Ranjit Kumar Sahu. (April 2011). ‘**Predict the onset of diabetes disease using Artificial Neural Network (ANN)**’. *International Journal of Computer Science & Emerging Technologies* (E-ISSN: 2044-6004). Volume 2, Issue 2, pp: 303-311.
- [PREVOTET\_2003] J.-C. Prevotet, B. Denby, P. Garda, B. Granado and C. Kiesling. (2003). ‘**Moving NN Triggers to Level-1 at LHC Rates**’. *Nuclear Inst. and Methods in Physics Research*, Elsevier.

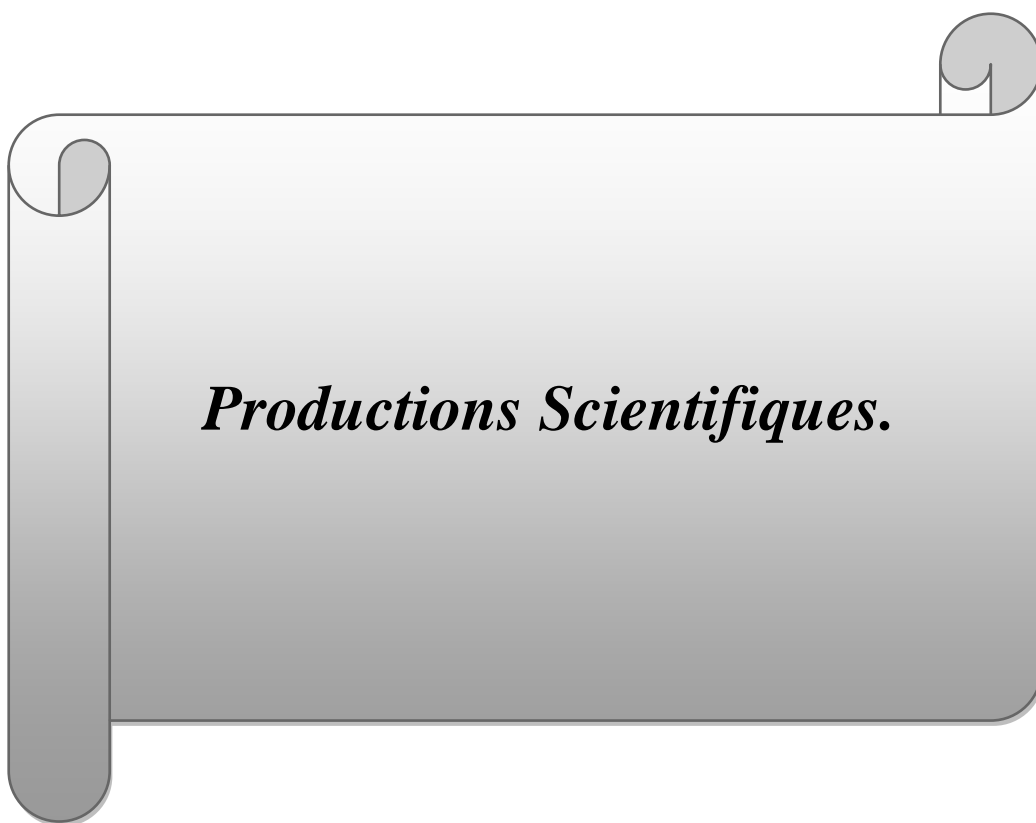
- [RAFID\_2007] Rafid Ahmed Khalil. (2 Sep. 2007). **‘Hardware Implementation of Backpropagation Neural Networks on Field programmable Gate Array (FPGA)’**. *University of Mosul, College of Engineering, Mosul, Iraq*.
- [RAI\_2013] Hari Mohan Rai, Anurag Trivedi, Shailja Shukla. (November 2013). **‘ECG signal processing for abnormalities detection using multi-resolution wavelet transform and Artificial Neural Network classifier’**. *Measurement*. Volume 46, Issue 9, Pages 3238–3246.
- [RAPHAËL\_2010] Raphaël Mourad, Christine Sinoquet, Philippe Leray. (2010). **‘Learning hierarchical Bayesian networks for genome-wide association studies’**. *Proceedings of COMPSTAT’2010*. Pages : 549-556. Éditeur : Physica-Verlag HD.
- [RENE\_2006] René Quiniou, Marie-Odile Cordier, Elisa Fromont, François Portet. (2006). **‘Reconnaissance D’arythmies Cardiaques’**. *Interstices Explorez les sciences du numérique*.
- [ROSARIA\_1998] S. Rosaria, M. Carlo. (1998). **‘Artificial neural networks for automatic ECG analysis’**. *IEEE Trans. Signal Processing*, 1417–1425.
- [ROUX\_1985] Maurice ROUX. (1985). **‘ALGORITHMES DE CLASSIFICATION’**. Editions Masson, Paris.
- [RUMEL\_1986] D.E. Rumelhart, G.E. Hinton & R.J. Williams. (1986). **‘Learning Internal Representations By Error Propagation’**. *Parallel Distributed Processing*, 1(8), pp 319-362.
- [SAHIN\_2006] Suhap Sahin, Yasar Becerikli and Suleyman Yazici. (2006). **‘Neural Network Implementation in Hardware Using FPGAs’**. *ICONIP 2006, Part III, LNCS 4234*, pp. 1105 – 1112, 2006. Springer-Verlag Berlin Heidelberg.
- [SALEHEEN\_2010] M. U. Saleheen, H. Alemzadeh, A. M. Cheriyan, Z. Kalbarczyk, and R. K. Iyer. (Oct. 2010). **‘An efficient embedded hardware for high accuracy detection of epileptic seizures’**. *In Proceedings of 3rd International Conference on Biomedical Engineering and Informatics*, vol. 5, pp. 1889-1896.
- [SALONI\_2015] Saloni, R. K. Sharma, Anil K. Gupta. (May 2015). **‘Voice Analysis for Telediagnosis of Parkinson Disease Using Artificial Neural Networks and Support Vector Machines’**. *I.J. Intelligent Systems and Applications*, 06, 41-47.
- [SEKKAL\_2011] M. Sekkal, M. CHIKH and N. SETTOUTI. (2011). **‘Evolving neural networks using a genetic algorithm for heartbeat classification’**. *Journal of Medical Engineering & Technology*, Volume 35, Issue 5.

- [SEKKAL\_2012] Sekkal,M. and Chikh,M.A. (2012). **‘NEURO-genetic approach to classification of cardiaque arrhythmias’**. *Journal of Mechanics in Medicine and Biology*, Vol. 12, No. 1.
- [SELVATHI\_2016] Selvathi, D. Nayagam, R. Deiva. (2016). **‘FPGA implementation of on-chip ANN for breast cancer diagnosis’**. *Intelligent Decision Technologies*, vol. Preprint, no. Preprint, pp. 1-12.
- [SETTOUTI\_2012] Nesma Settouti, M. Amine Chikh, Meryem Saidi. (2012). **‘Generating fuzzy rules for constructing interpretable classifier of diabetes disease’**. *Australasian Physical & Engineering Sciences in Medicine*. ISSN 0158-9938.
- [SINGH\_2016] Bikesh Kumar Singh, Kesari Verma, A.S. Thoke. (30 December 2016). **‘Fuzzy cluster based neural network classifier for classifying breast tumors in ultrasound images’**. *Expert Systems with Applications*. Volume 66, Pages 114–123.
- [SOMMELET\_2009] Danièle Sommelet, Jacqueline Clavel, Brigitte Lacour. (2009) **‘Epidémiologie des cancers de l’enfant’**. Publisher : Springer-Verlag Paris, Softcover ISBN : 978-2-287-78336-4.
- [SPARTAN\_2013] Spartan-3E FPGA Family Data Sheet, DS312. (July 19, 2013). *Product Specification*. xilinx
- [SPECHT\_1990] Specht, D. F. (1990). **‘Probabilistic neural networks’**. *Neural networks*, 3(1):109–118
- [SRIDHAR\_2013] D. Sridhar, IV. Murali Krishna. (2013). **‘Brain Tumor Classification using Discrete Cosine Transform and Probabilistic Neural Network’**. *Published in: Signal Processing Image Processing & Pattern Recognition (ICSIPR), 2013 International Conference on*. Date Added to IEEE Xplore: 15 April 2013.
- [TARIGHT\_1999] Y. TARIGHT. (19 Janvier 1999). **‘Contribution A L’Analyse De La Pollution Atmosphérique Chronique Ou Accidentelle : Concept De Nez Electronique Ambulatoire’**. Thèse De Doctorat, Rouen.
- [UBEYLI\_2010] Ubeyli ED (September 2010). **‘Automatic diagnosis of diabetes using adaptive neuro-fuzzy inference systems’**. *Expert System*. 27(4):259–266. Volume 27, Issue 4. Pages 259–266.
- [VANITHA\_2011] Vanitha.L, Venmathi.A.R. (2011). **‘Classification of Medical Images Using Support Vector Machine’**. *International Conference on Information and Network Technology IACSIT Press*, Singapore 63 IPCSIT vol.4.
- [WANG\_2013] Jeen-Shing Wang, Wei-Chun Chiang, Yu-Liang Hsu, Ya-Ting C. Yang. (20 September 2013). **‘ECG arrhythmia**



- classification using a probabilistic neural network with a feature reduction method**'. *Neurocomputing, Advanced Theory and Methodology in Intelligent Computing*. Volume 116, Pages 38–45.
- [WEBER\_2000] Jacques WEBER. (Avril 2000). **'Les Circuits Logiques Programmables'**. *Département d'électronique*. CACHAN.
- [WEBER\_2007] Jacques WEBER & Sébastien MOUTAULT & Maurice MEAUDRE. (2007). **'Le Langage VHDL, Du Langage Au Circuit, Du Circuit Au Langage'**. Dunod, Paris, ISBN 978-2-10-050191-5.
- [WHO\_1999] (1999). **'Definition, diagnosis and classification of diabetes mellitus and its complications. Part 1: Diagnosis and classification of diabetes mellitus'**. *Organisation mondiale de la Santé*, Genève. Report Number: WHO/NCD/NCS/99.2.
- [WIKIPEDIA\_09/2016] [https://fr.wikipedia.org/wiki/Carte\\_auto\\_adaptative](https://fr.wikipedia.org/wiki/Carte_auto_adaptative).
- [WIKIPEDIA\_10/2016] [https://en.wikipedia.org/wiki/Probabilistic\\_neural\\_network](https://en.wikipedia.org/wiki/Probabilistic_neural_network).
- [WIKIPEDIA\_2014] <https://fr.wikipedia.org/wiki/Verilog>.
- [WIKIPEDIA\_2015] <https://fr.wikipedia.org/wiki/SystemC>.
- [XILINX\_1999] Product Specification. (May 14, 1999). **'XC4000E and XC4000X Series Field Programmable Gate Arrays'**. Version 1.6 [https://www.xilinx.com/support/documentation/data\\_sheets/4000](https://www.xilinx.com/support/documentation/data_sheets/4000).
- [XILINX\_2000] Product Specification, (2000). Xilinx.
- [XILINX\_2008] Product Specification. (June 13, 2008). **'Spartan-II FPGA Family Data Sheet DS001'**. [https://www.xilinx.com/support/documentation/data\\_sheets/ds001](https://www.xilinx.com/support/documentation/data_sheets/ds001).
- [ZAINUDDIN\_2012] Zarita Zainuddin ,Lai Kee Huong, Ong Pauline. (2012). **'On the Use of Wavelet Neural Networks in the Task of Epileptic Seizure Detection from Electroencephalography Signals'**. *Procedia Computer Science. Proceedings of the 3rd International Conference on Computational Systems-Biology and Bioinformatics*. Volume 11, Pages 149-159.
- [ZHANG\_1992] Q. Zhang, A. Beneveniste. (Novembre 1992). **'Wavelet Networks'**. *IEEE Transactions on Neural Networks* 3(6) : 889-898.
- [ZHOU\_2003] J. Zhou. (2003). **'Automatic Detection of Premature Ventricular Contraction Using Quantum Neural Networks'**. *Proceedings of 3rd IEEE Symposium on BioInformatics and BioEngineering*. *IEEE Computer Society*, 3, 169-173.

- [ZIDELMAL\_2012] Z. Zidelmal, A. Amirou, A. Bellouchrani. (2012). '**Heartbeat classification using Support Vector Machines (SVMs) with an embedded rejet option**'. *International Journal of Pattern Recognition and Artificial Intelligences (IJPRAI)*, Vol. 26, No. 01, 1250001.
- [ZIGHED\_2000] Djamel A. Zighed, Ricco Rakotomalala. (2000). '**Graphes D'induction (Apprentissage Et Data Mining)**'. Edition hermès. P25-27.
- [ZITOUNI\_2014] Pr Messaoud Zitouni, Mr F. AMOKRANE, Dr F. CHERF-BOUZIDA. (Octobre 2014). '**Plan National Cancer 2015-2019 : nouvelle vision stratégique centrée sur le malade**'.



***Productions Scientifiques.***

## **Publications.**

- ✚ BABA HAMED, H. BECHAR, M.A. CHIKH. (March 2016). **'Implementation of a Neural Classifier on a FPGA-Based Reconfigurable of Cardiac Arrhythmias'**. *EEA: Electrotehnica, Electronica, Automatica*, Vol.64, Nr.1, PP.177-186. ISSN: 1582-5175.

## **Communications Internationales.**

- ✚ BABA HAMED Amel, BECHAR Hassane, CHIKH Mohammed Amine. (15-16 octobre 2014). **'Conception Matérielle D'un Classifieur Neuronal En Technologie FPGA En Vue D'une Classification Médicale'**. *Biomedical Engineering International Conference (BIOMEIC'14)*. Tlemcen.

## **Communications Nationales.**

- ✚ BABA HAMED Amel, BECHAR Hassane, CHIKH Mohammed Amine. (23 Mai 2013). **'Implémentation d'un Classifieur Neuronal dans un FPGA pour la Reconnaissance du Diabète'**. *3<sup>ème</sup> Journée Doctorale de Génie Biomédical (JD-GBM'13)*. Tlemcen.
- ✚ BABA HAMED Amel, BECHAR Hassane, CHIKH Mohammed Amine. (15 Mai 2014). **'Implémentation d'un Réseau de Neurones sur un circuit reconfigurable FPGA en Vue D'une Classification des Données Biomédicales'**. *4<sup>ème</sup> Journée Doctorale de Génie Biomédical (JD-GBM'2014)*. Tlemcen.
- ✚ BABA HAMED Amel, BECHAR Hassane, CHIKH Mohammed Amine. (11 Juin 2015). **'Détection des Dépassements de Capacité lors de la Conception Matérielle du Classifieur Neuronal en Technologie FPGA'**. *5<sup>ème</sup> Journée Doctorale de Génie Biomédical (JD-GBM'15)*. Tlemcen.

## **Productions Pédagogiques.**

- ✚ BABA HAMED Amel, BECHAR Hassane. (Juin 2014). **'Description Matérielle Des Circuits Numériques En Langage VHDL'**. *Manuel des Travaux Pratiques*. Editions Internes de la Faculté de Technologie de l'Université de Tlemcen.

