

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Electronique
Laboratoire signal et communications



Thèse de Doctorat en Sciences En Electronique

Présentée par :

Faroudja ABID

Magister en Electronique de l'ENP

Intitulé

Implémentation sur FPGA d'un système de classification des chromosomes humains

Soutenue publiquement le 29/04/2018 devant le jury composé de :

Président :	Mme. Mhania GUERTI	Professeur	ENP
Rapporteur :	Mme. Latifa HAMAMI	Professeur	ENP
Examineurs :	Mme. Rachida TOUHAMI	Professeur	USTHB
	M. El-Bay BOURENNANE	Professeur	Université de BOURGOGNE
	M. Mohamed BOUAMAR	Professeur	ESDAT
	M. Chérif LARBES	Professeur	ENP

ENP 2018

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Electronique
Laboratoire signal et communications



Thèse de Doctorat en Sciences En Electronique

Présentée par :

Faroudja ABID

Magister en Electronique de l'ENP

Intitulé

Implémentation sur FPGA d'un système de classification des chromosomes humains

Soutenue publiquement le 29/04/2018 devant le jury composé de :

Président :	Mme. Mhania GUERTI	Professeur	ENP
Rapporteur :	Mme. Latifa HAMAMI	Professeur	ENP
Examineurs :	Mme. Rachida TOUHAMI	Professeur	USTHB
	M. El-Bay BOURENNANE	Professeur	Université de BOURGOGNE
	M. Mohamed BOUAMAR	Professeur	ESDAT
	M. Chérif LARBES	Professeur	ENP

ENP 2018

Remerciements

J'adresse mes remerciements à ma directrice de thèse le Pr. Latifa HAMAMI qui m'a confié ce travail et aidé à sa réalisation et pour son soutien et ses encouragements.

Je remercie également la présidente du jury, Madame Mhania GUERTI, Professeur à l'Ecole Nationale Polytechnique d'Alger qui m'a fait l'honneur de présider le jury de cette thèse.

J'adresse mes sincères remerciements aux membres du jury, Madame. Rachida TOUHAMI, Professeur à l'Université des Sciences et de Technologie Houari Boumediene d'Alger, Monsieur El-Bay BOURENNANE, Professeur à l'université de Bourgogne, Monsieur Mohamed BOUAMAR, Professeur à l'Ecole Supérieure de la Défense Aérienne du Territoire (ESDAT) et Monsieur Larbes Chérif, Professeur à l'Ecole Nationale Polytechnique d'Alger qui ont accepté d'examiner et d'évaluer ce travail de thèse.

ملخص

استخدمت الشبكات العصبية الاصطناعية لتصنيف الكروموزومات و كشف الانحرافات الكروموزومية بسبب بنيتها المتوازية التي تقلل من التعقيد ووقت الحساب، إلا أن الأنظمة الموجودة تعتمد على برامج تعمل على جهاز الكمبيوتر الذي يحول خاصية التوازي في الشبكات العصبية إلى عمليات تسلسلية ويقلل من سرعتها الحسابية.

يهدف هذا العمل أساسا إلى تنفيذ وانجاز نظام على شريحة الكترونية SoC-AKS لتصنيف الكروموزومات عند الإنسان. النموذج الأول من نوعه الذي تم تصميمه على شريحة FPGA يتركز على الشبكات العصبية و يسمح بتحقيق نظام محمول ومصغر باستهلاك محدود للطاقة من جهة، ومن جهة أخرى يسمح بالمساعدة على كشف الانحرافات الكروموزومية بطريقة أوتوماتيكية.

الكلمات الرئيسية : الكروموزومات, الشبكات العصبية, شريحة الكترونية FPGA, تصنيف الكروموزومات.

Abstract

Artificial neural networks (ANNs) have been widely used in chromosome classification due to their parallelism that reduces the computational complexity and time. However, existing classifiers are software-based, running on a computer that transforms the parallelism features of the ANNs into serial operations, thus significantly reducing their computing power. To efficiently address the above issue due to software implementation, we propose a Field-Programmable Gate Array-based System on Chip (SoC) architecture for human chromosome classification. The hardware implementation of such system can achieve the parallelism inherent to ANNs while reducing the power consumption and circuit size, thus the cost of designing such a system is reduced. The proposed SoC-AKS (System on Chip-Automatic karyotyping System) aims to realize a portable and automatic system that helps the cytologists in karyotyping process and the genetic syndrome diagnosis while reducing time, effort and cost.

Key words: ANN, Human chromosome, FPGA, Karyotyping, SoC

Résumé

Notre travail consiste en l'implémentation hardware d'un système de classification des chromosomes humains basé sur les réseaux de neurones. Le SoC-AKS (*System on Chip-Automatic karyotyping System*) proposé cible un support hardware de type plateforme FPGA (*Field-Programmable Gate Array*). L'intérêt de la solution hardware proposée est de garantir, d'une part, la portabilité, la faible consommation de puissance et la miniaturisation d'un tel système; et d'autre part, obtenir de meilleures performances de fonctionnement en exploitant le parallélisme inhérent aux ANNs, via une implémentation sur une architecture reconfigurable de type FPGA en réduisant le temps de calcul lié à l'exécution software sur PC. Le SoC-AKS proposé vise à réaliser un système portable qui permet l'établissement du caryotype et le diagnostic de syndromes génétiques.

Mots clés : ANN, Chromosome humain, Caryotype, FPGA, SoC.

TABLE DES MATIERES

Liste des figures

Liste des abréviations

Liste des tableaux

Introduction Générale.....12

Chapitre 1

Etat de l'art

1.1 Introduction	17
1.2 Techniques et méthodes d'extraction des caractéristiques des Chromosomes	17
1.2.1 Technique de localisation du centromère.....	18
1.2.2 Extraction de l'axe médian du chromosome.....	20
1.3 Méthodes de classification des chromosomes basées sur les réseaux de neurones artificiels (ANNs)	22
1.3.1 Modèles basés sur le réseau perceptron multicouches.....	22
1.3.2 Autres modèles d'ANN pour la classification des chromosomes.....	22
1.4 Classificateurs à base d'autres algorithmes	27
1.4.1 Classificateurs à base de l'algorithme SVM.....	27
1.4.2 Classificateurs additionnel	28
1.5 Discussion et conclusions	28

Chapitre 2

Les systèmes sur puce

2.1 Introduction et définition	31
2.2 Architecture	32
2.3 Les méthodologies de conception SoC /Approche SoC	32
2.3.1 Flot de conception SoC	33
2.3.2 Approche réutilisation d'IPs	35
2.3.3 Implémentation sur FPGA	36
2.3.3.1 Avantagesetinconvénients des circuits FPGAs.....	36
2.3.3.2 Flot de conception FPGA.....	36
2.3.3.3 Les vérifications dans les FPGAs	38
2.3.4 Implémentation sur ASICs	39
2.3.4.1 Les ASICs	39

2.3.4.2 Avantages et inconvénients des ASICs.....	41
2.3.4.3 Flot de conception ASIC.....	42
2.3.4.4 Les vérifications dans les ASICs.....	42
2.4 La synthèse haut niveau (HLS).....	44
2.5 Conclusion.....	46

Chapitre 3 Approche Méthodologique de développement du SoC-AKS

3.1 Introduction.....	47
3.2 Système AKS proposé.....	47
3.2.1 Concept d'un système de classification des chromosomes.....	47
3.2.2 Description du AKS proposé.....	48
3.3. Approche méthodologique de développement.....	50
3.3.1 Le diagramme en Y.....	50
3.3.2.1 Le domaine comportemental.....	51
3.3.2.2 Le domaine structurel.....	51
3.3.2.3 Le domaine physique.....	51
3.4 Approche de conception du système de classification des chromosomes humains sur FPGA.....	52
3.4.1 Choix du modèle ANN.....	52
3.4.2 Choix du support hardware.....	53
3.4.3 Choix du langage.....	53
3.4.4 Définition du modèle architectural du SoC-AKS.....	54
3.5 Définition des modèles architecturaux modélisant les différents sous-systèmes.....	56
3.5.1 Définition du modèle architectural modélisant la phase de traitement d'image.....	56
3.5.2 Définition du modèle architectural modélisant la phase extraction des caractéristiques.....	57
3.5.2.1 Méthodologie adoptée pour l'extraction des caractéristiques.....	57
3.5.2.2 Calcul des caractéristiques.....	58
3.5.3 Définition du modèle architectural modélisant la phase de classification.....	60
3.5.3.1 Description de l'algorithme de Kohonen.....	60
3.5.3.2 Flot de conception du sous-système classification.....	61
3.6 Conclusion.....	64

Chapitre 4 Modélisation et implémentation hardware

4.1 Introduction.....	65
4.2 Architecture globale.....	65
4.2.1 Diagramme bloc du système proposé.....	65
4.2.2 Présentation de l'architecture hardware du sous-système classification.....	67
4.3 Implémentation des deux phases de l'algorithme de Kohonen.....	68

4.3.1 Implémentation de la phase d'apprentissage.....	68
4.3.1.1 Bloc mémorisation des paramètres.....	69
4.3.1.2 Bloc détermination du BMU	69
4.3.1.3 Bloc mise à jour des poids synaptiques	73
4.3.2 Implémentation de la phase classification.....	75
4.4 Présentation de l'architecture hardware du sou-système extraction des caractéristiques.....	76
4.4.1 Calcul des caractéristiques	77
4.4.1.1 Architecture du module PVH et localisation du centromère.....	77
4.4.1.2 Calcul de la longueur du chromosome	79
4.4.1.3 Calcul de l'indice centromérique (IC).....	80
4.4.1.4 Calcul des densités de profil	81
4.5 Conclusion.....	83

Chapitre 5

Prototypage et résultats

5.1 Introduction	84
5.2 Prototypage	85
5.2.1 Résultats de synthèse et implémentation.....	85
5.2.2 Résultats en terme de puissance consommée.....	88
5.3 Evaluation des performances	89
5.3.1 Taux de reconnaissance.....	89
5.3.2 Evaluation des performances en terme de MCPS et MCUPS.....	90
5.4 Conclusion.....	92
Conclusions et perspectives.....	94
Bibliographie	97
Annexe A : Généralités sur les chromosomes.....	104
Annexe B : Algorithme de Kohonen.....	111

LISTE DES FIGURES

Figure 2.1 : Architecture d'un système sur puce	32
Figure 2.2 : Flot de conception SoC.....	34
Figure 2.2 (a) : Conception générique.....	34
Figure 2.2 (b) : Conception Co-design.....	34
Figure 2.3 : Le flot de conception FPGA	37
Figure 2.4 (a): Flot de conception FPGA de Xilinx.....	38
Figure 2.4 (b): Détails des fichiers dans chaque processus.....	38
Figure 2.5 : Types d'approches de conception ASIC	40
Figure 2.6 : Flot de conception ASIC	43
Figure 2.7 : Approche HLS	44
Figure 2.8 : Flot de conception Vivado HLS	45
Figure 2.9 : Flot de conception Vivado Stratus.....	45
Figure 3.1 : Les principales étapes d'un AKS	48
Figure 3.2 : Schéma blocs d'un système proposé pour la classification des chromosomes	50
Figure 3.3: Le diagramme en Y	51
Figure 3.4: Flot de conception du AKS	55
Figure 3.5: Flot de conception de la phase traitement de l'image.....	56
Figure 3.6: Etapes de la méthodologie adoptée pour l'extraction des paramètres.....	58
Figure 3.7: La courbe des densités de profil du chromosome.....	59
Figure 3.8 : Carte de Kohonen	61
Figure 3.9 : Flot de conception du système classification à base de l'algorithme de Kohonen	63
Figure 4.1 : Schéma bloc du SoC-AKS	66
Figure 4.2 : Diagramme bloc de l'architecture du sous-système classification.....	67
Figure 4.3 : Schéma bloc de l'architecture de la phase d'apprentissage.....	69

Figure 4.4 : Architecture du bloc détermination de BMU	70
Figure 4.5 : Architecture interne (schématique) du module calcul de la distance du 1 ^{er} neurone.....	71
Figure 4.6 : Comparaison de distance pour 8 neurones	72
Figure 4.7 : Architecture interne (schématique) du module sélection du neurone gagnant.....	72
Figure 4.8 Architecture interne (schématique) du module BMU.....	78
Figure 4.9 : Architecture interne (schématique) du module de mise à jour de poids du jéme neurone	75
Figure 4.10 : Schéma bloc de l'architecture de la phase de classification	76
Figure 4.11 : Schéma synoptique de l'architecture hardware du sous-système extraction des caractéristiques.....	76
Figure 4.12 : Architecture du module PVH et calcul du minimum global	78
Figure 4.13 : Centromère qui sépare le chromosome en deux bras	79
Figure 4.14: Architecture interne (schématique) du module calcul des points du vecteur PVH.....	79
Figure 4.15 : Architecture du module calcul de la longueur.....	80
Figure 4.16 : Architecture interne du module calcul de la longueur du chromosome	81
Figure 4. 17 : Architecture du Module calcul de DP	82
Figure 4. 18 : Architecture interne (schématique) du module calcul de la DP	82
Figure 5.1 Résultats de la classification des chromosomes affichés sur l'écran LCD de l'FPGA.....	85
Figure 5.2 : Layout de l'architecture du sous-système classification.....	88

LISTE DES TABLEAUX

Tableau 1.1: Résumé de quelques études concernant les méthodes d'extraction des caractéristiques.....	21
Tableau 1.2 : Résumé de quelques études concernant les systèmes de classification des chromosomes basés sur les ANNs.....	26
Tableau 2.1 : Comparaison des performances des différentes technologies.....	41
Tableau 5.1 :Résultats de synthèse Virtex 6	86
Tableau 5.2: Résultats de synthèse Virtex 7	86
Tableau 5.3: Résultats de synthèse Kintex 7.....	86
Tableau 5.4 : Résultats de synthèse Spartan 3	86
Tableau 5.5 : Résultats de synthèse pour une Virtex5 (XC5VLX50-1FF676).....	87
Tableau 5.6 : Résultats de synthèse pour différents circuits FPGA (résumé).....	87
Tableau 5.7 : Puissance consommée pour différents circuits FPGAs.....	88
Tableau 5.8 : Comparaison des performances en termes de taux de classification ...	89
Tableau 5.9: Comparaison des performances en termes de MCPS et MCUPS	91
Tableau 5.10: Comparaison des performances en termes de MCPS sur divers plateformes FPGAs de Xilinx	92

LISTE DES ABREVIATIONS

ADN	Acide Désoxyribo-Nucléique
AKS	Automatic Karyotyping System
AMS	Austria Mikro System
ANN	Artificial Neuron Network
AODA	Automatic Object Detection Algorithm
ASIC	Application Specific Integrated Circuit)
ATPG	Automatic Test Pattern Generation
BIST	Built-In Self-Test
BMU	Best Matching Unit
BP	Back-Propagation
CAO	Conception Assistée par Ordinateur
CFHN	Classifier Fuzzy Hopfield neural Network
CGH	Comparative Genomic Hybridization
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Device
DAPI	4-6-Diamidino-2-Phenylindole
DC	Design Compiler
DFT	Design For Test logic insertion
DP	Density Profile
DPLL	Digital Phase-Locked Loop
DSP	Digital Signal Processor
DWT	Discrete Wavelet Transform
EDA	Electronic Design Automation
FISH	Fluorescence In-Situ Hybridization
FPGA	Field-Programmable Gate Array
GUI	Graphical User Interface
HDL	Hardware Description Language
HLS	High Level Synthesis
IC	Indice Centromérique
IC	Integrated Circuit
IOB	In Out Block
IP	Intellectual Properties
ISE	Integrated Software Environnement
LUT	Look-Up-Tables
MAT	Medial Axis Transform
MCPS	Million-Connection-Per-Second
MCUPS	Million-Connection-Update-Per-Second
M-FISH	Multiplex-FISH

ML	Maximum Likelihood
MLP	Multi-Layer Perceptron
NBC	Naïve Bayesian Classifier
NRE	Non Requiring Engineering
OSA	Object Skeletonizing Algorithm
PDK	Process Design Kit
PWL	PieceWise Linear
PV	Projection Vector
RAM	Random Access Memory
RBC	Rule Based Classifier
RTL	Register Transfer Level
SoC	System on Chip
SoC- AKS	System on Chip-Automatic Karyotyping System
SOM	Self-Organizing Map
STA	Static Timing Analysis
SVM	Support Vector Machine
TTM	Time To Market
UCF	User Constraints File
VLSI	Very-Large-Scale Integration
WNN	Wavelet Neural Network

Introduction Générale

La corrélation entre les maladies génétiques et les aberrations chromosomiques a été établie depuis la découverte du nombre des chromosomes d'une cellule humaine par *Tjio et Levan* [1] en 1956. Ces anomalies ont été décrites entre 1959 et 1965, elles sont le résultat d'un changement ou d'une réorganisation du nombre des chromosomes ou de leurs structures. Ainsi, l'analyse de la morphologie du chromosome et le processus d'établissement du caryotype (une représentation standard des 23 paires de chromosomes) sont des procédures importantes pour la détection de ces aberrations et les syndromes cliniques qui en découlent [2], comme par exemple le diagnostic des cancers. Cependant ces processus sont réalisés manuellement dans la plupart des laboratoires de cytogénétique et nécessitent une intervention des cytogénéticiens dans toutes les étapes. En conséquence l'établissement du caryotype s'avère lent et donc coûteux.

Depuis 1980 de nombreux systèmes d'aide au diagnostic [3-9] ont été développés pour l'automatisation de la procédure de classification des chromosomes, qui est une étape prépondérante dans l'automatisation de la routine d'établissement du caryotype et la détection des anomalies chromosomiques. En particulier les modèles basés sur les approches réseaux de neurones artificiels (ANNs) [5, 10].

Les modèles ANNs ont été adoptés vu leur capacité de traitement relative à leurs architectures parallèles. La plupart des travaux effectués dans ce domaine consistent en des implémentations softwares s'exécutant sur un ordinateur conventionnel. Toutefois, une implémentation software transforme le parallélisme des ANNs en opérations séquentielles; ce qui réduit leur puissance de calcul. Compte tenu des travaux revus dans la littérature, il n'y a aucune implémentation sur support hardware pour les systèmes de classification des chromosomes. Les systèmes existants sont basés sur des implémentations softwares, nous supposons qu'une implémentation hardware d'un tel système peut réaliser le parallélisme inhérent aux réseaux de neurones.

D'autre part, les systèmes automatisés disponibles dans le commerce pour l'établissement du caryotype sont semi-automatiques et demeurent coûteux. Nous proposons une solution qui réalise un système sur puce pour la classification des chromosomes humains, tout en réduisant la consommation de puissance et la taille du circuit. Ainsi, le coût de conception d'un tel système est réduit.

En effet, notre travail s'inscrit dans la thématique des systèmes d'aide au diagnostic; nous proposons à travers ce travail de thèse un système sur puce (SoC) pour la classification des chromosomes humains. L'approche méthodologique que nous proposons est développée dans le but d'envisager une démarche permettant l'implémentation d'un système de classification des chromosomes humains sur un support hardware de type FPGA (*Field-Programmable Gate Array*), en particulier les parties principales de ce système, à savoir le sous-système classification et le sous-système extraction des paramètres (caractéristiques).

Une implémentation typiquement hardware, dont l'intérêt est la réalisation d'un système portable et rapide en exploitant la rapidité de traitement et le parallélisme d'un circuit FPGA, est proposée. Cette conception hardware permet l'automatisation de la tâche de classification tout en exploitant le parallélisme du réseau de neurones. Le SoC-AKS (*System on Chip-Automatic Karyotyping System*) proposé vise à réaliser un système hardware qui permet l'établissement du caryotype et le diagnostic des syndromes génétiques avec une intervention humaine minimale.

Concevoir des modules entièrement parallèles peut se faire sur des circuits de type ASICs (*Application Specific Integrated Circuits*) et VLSI (*Very-Large-Scale Integration*), cependant le développement de tels circuits est un processus coûteux et lent. Le choix d'une plateforme FPGA est justifié par le parallélisme, un cycle de développement relativement court et la rapidité de traitement qu'assurent ces architectures reconfigurables. Les motivations pour un système hardware de classification automatique des chromosomes; basé sur une plateforme FPGA sont principalement:

1. Le parallélisme

Une implémentation hardware sur FPGA d'un tel système exploite le parallélisme inhérent aux réseaux de neurones et réduit le coût d'une conception, qui peut être testée et évaluée rapidement, permettant un temps de développement relativement court.

2. Temps de traitement réduit

Une amélioration majeure avec une implémentation hardware d'un tel classificateur de chromosomes est dans le temps d'apprentissage, c'est-à-dire dans l'amélioration du temps de convergence du réseau, qui est un facteur important. En effet, le temps de convergence des classificateurs ANNs basés sur des implémentations softwares est de l'ordre de quelques secondes, alors que le temps de convergence du classificateur ANN hardware proposé est de l'ordre de microsecondes (μs).

3. Réduction de la puissance consommée

La consommation de puissance d'un SoC pour la classification des chromosomes humains basé sur FPGA est faible, vu que la solution est embarquée.

4. Miniaturisation

Le SoC basé sur FPGA, permet une miniaturisation du système de classification des chromosomes et réduit considérablement l'espace utilisé par le PC ou les équipements conventionnels dédiés pour un tel système.

Notre principal objectif est l'implémentation hardware de la partie classification constituant la phase fondamentale dans un système de classification des chromosomes. Ce dernier est basé dans notre cas sur l'approche ANN, le réseau adopté étant une combinaison de deux ANNs de type Kohonen. Plusieurs travaux [11-13] ont été effectués dans notre laboratoire pour la classification des chromosomes avec des implémentations softwares basées notamment, sur le modèle MLP (*Multi-Layer Perceptron*), cependant notre objectif est une implémentation hardware qui doit répondre à des contraintes de surface utilisée, de consommation de puissance, et de vitesse de calcul; c'est pourquoi le choix est fait pour le modèle de Kohonen qui requière moins de temps de calcul dans les phases d'apprentissage et de classification, et donc moins de surface sur un support hardware.

La topologie choisie pour le classificateur multi-réseau découle d'un travail déjà effectué au niveau de notre laboratoire [11]. Le premier réseau classifie les chromosomes par leurs caractéristiques géométriques (Longueur (L) et Indice Centromérique (IC)) et le deuxième réseau les classifie en fonction de leurs densités de profil (DPs).

Le système proposé comprend, principalement, trois sous-systèmes à savoir, le sous-système traitement des images métaphasiques, le sous-système extraction des caractéristiques et le sous-système classification. La phase de traitement de l'image inclut le prétraitement de l'image et la segmentation. Le prétraitement est élaboré pour faciliter la segmentation et l'identification des chromosomes. La segmentation constitue la deuxième étape dans la phase de traitement de l'image, après binarisation, un autre algorithme est appliqué pour individualiser les chromosomes. A l'issue de cette phase les images des chromosomes sont prêtes pour le processus de squelettisation et l'extraction des paramètres.

Le second sous-système est donc, le sous-système extraction des paramètres. La squelettisation de l'image de chromosome, pour l'extraction de son axe médian est effectuée en premier lieu, avant l'extraction des caractéristiques. Le résultat de traitement de ce sous-système est un vecteur de paramètres qui inclut la longueur du chromosome, l'indice centromérique et la densité de profil.

Après étude des différentes techniques d'extraction des caractéristiques, nous avons opté pour une association de deux techniques, à savoir la transformation de l'axe médian (MAT : *Medial Axis Transform*) [14] et la projection vectorielle (PV) horizontale [15]. Après squelettisation, l'axe médian est utilisé pour le calcul des longueurs des chromosomes et les densités de profil. Nous avons choisi la technique PV horizontale pour l'identification du centromère (représenté par la région de constriction dans la structure du chromosome) et le calcul, par la suite, de l'indice centromérique. Le centromère est considéré comme le minimum global dans la projection vectorielle horizontale.

Le dernier sous-système est le sous-système classification basé sur deux réseaux neuronaux de type Kohonen, il représente la partie principale du SoC, cette partie a été conçue et implémentée sur une plateforme FPGA Virtex5. Chaque sous-système est modélisé par une architecture hardware dans le SoC-AKS.

Ce manuscrit est structuré en cinq chapitres :

Dans le premier chapitre nous exposerons une étude de l'état de l'art sur l'extraction des caractéristiques des chromosomes, et les différentes méthodes de classification des chromosomes basées sur les ANNs. Nous aborderons également, les classificateurs basés sur l'approche SVM (Support Vector Machine). Nous concluons

ce chapitre par une discussion présentant une analyse et comparaison des différents travaux passés en revue.

Le deuxième chapitre, présente les différentes approches et flots de conception SoC, sur lesquels est basée notre approche de conception. L'approche méthodologique proposée pour une implémentation hardware d'un système automatique de caryotype sur support hardware de type FPGA, fera l'objet du troisième chapitre.

Le quatrième chapitre traitera de l'implémentation proprement dite. Nous allons décrire la solution architecturale proposée pour l'implémentation du SoC-AKS, en détaillant l'ensemble des blocs fonctionnels qu'il contient. Nous nous focaliserons sur les principales parties de ce système à savoir, l'extraction des caractéristiques et classification.

Nous exposerons dans le dernier chapitre, les résultats de prototypage du sous-système finalisé, à savoir le sous-système classification. Nous présenterons, également les résultats d'une évaluation des performances du classificateur proposé en termes de taux de classification, de *Million-Connection-per-second* (MCPS) et de *Million-Connection-Update-per-second* (MCUPS). Nous concluons ce manuscrit par des conclusions et perspectives.

1

Etat de l'art

1.1 Introduction

Plusieurs travaux [3-9] portant sur les systèmes automatiques d'analyse et de classification des chromosomes ont été rapportés dans la littérature, ce domaine de recherche reste toujours un axe de recherche d'actualité.

Les techniques et méthodes d'extraction des caractéristiques des chromosomes ainsi que les différentes méthodes de classification des chromosomes basées sur les réseaux de neurones artificiels seront passées en revue dans ce chapitre. Nous aborderons également d'autres méthodes et approches, outre les ANNs, pour la classification des chromosomes humains, en particulier les SVMs (*Support Vector Machine*) qui ont fait l'objet de récents travaux [16-18].

1.2 Techniques et méthodes d'extraction des caractéristiques des chromosomes

Les études ayant portées sur la classification des chromosomes ont, pour la plupart, utilisé en plus des caractéristiques géométriques du chromosome (longueur ou périmètre, position du centromère, indice centromérique), les caractéristiques

textuelles relatives aux bandes chromosomiques. En particulier, les densités de profil, qui sont les moyennes des intensités des pixels constituant chaque ligne perpendiculaire à l'axe médian du chromosome. Les concepts de base de la cytogénétique et les définitions relatives aux chromosomes et leur analyse sont donnés en annexe A.

1.2.1 Technique de localisation du centromère

Les caractéristiques géométriques les plus utilisées sont la longueur du chromosome et la position du centromère. La localisation du centromère a fait l'objet de plusieurs travaux [14, 15, 19, 20]; par exemple dans [20] le centromère et la polarité des chromosomes ont été déterminés en utilisant un algorithme d'amincissement employé pour identifier l'axe médian du chromosome et l'extraction, par la suite, de trois paramètres de profil, à savoir la densité de profil, le profil de forme (*shape profile*) et le profil de bande (*banding profile*). La base de données utilisée dans cette étude comprend des chromosomes au nombre de 2287, extraits de 50 cellules métaphasiques de la moelle osseuse (dont 26 sont normales et 24 présentant des anomalies) d'un sujet suspect de leucémie. Des taux de réussite de 91.4% et 97.4% ont été rapportés pour l'identification du centromère et la polarisation, respectivement. Toutefois *Wang et al.* affirment que leur système présente une limitation dans l'identification du centromère pour les chromosomes de petites tailles (17, 18 et 20), qui sont soit chevauchés ou très courbés.

Une autre technique pour l'identification du centromère a fait l'objet du travail exposé dans [15]. La technique en question est nommée projection vectorielle (PV : *projection vector*), elle est basée sur la projection vectorielle verticale (PVV), définie comme la somme des pixels de chaque colonne de l'image binarisée du chromosome, et la projection vectorielle horizontale (PVH), représentée par la somme des pixels de chaque ligne de l'image binarisée du chromosome. Le centromère est considéré comme le minimum global dans la projection vectorielle horizontale. Le taux de reconnaissance de la position du centromère est de 96% pour 87 chromosomes. La principale limitation de cette technique est dans le cas des chromosomes fortement courbés. Les mêmes auteurs ont proposé une autre méthode [21] basée sur la localisation automatique des points clés (appelés bornes limites (*landmarks*)) dans le squelette du chromosome; ces bornes sont les points extrêmes du squelette.

La base de données utilisée dans cette étude est composée de 219 chromosomes du groupe E. Ce dernier est l'un des groupes de Denver, dont la classification est basée sur les caractéristiques géométriques, où les 46 chromosomes sont réparties en 7 groupes (A (1-3), B(4,5), C(6-12, X), D(13,15), E(16-18), F(19,20), G (21-22, Y)) selon leurs tailles décroissantes.

Récemment, *Madian et al.* [19] ont présenté une étude comparative entre la technique de transformation de l'axe médian (MAT) et la projection vectorielle (PV), qui sont employées pour la détermination de la position du centromère. Un algorithme de squelettisation (*distance transform*) a été utilisé pour l'extraction de l'axe médian. Selon cette étude, la technique MAT est appropriée pour les chromosomes droits, seulement, et les performances de cet algorithme se dégradent pour les chromosomes courbés. Le centromère est déterminé pour 55 chromosomes; l'approche PV permet une meilleure détection du centromère, pour les chromosomes droits et courbés avec un taux de réussite de 90%. Tandis qu'un taux de réussite de 78% a été rapporté pour la technique MAT. Parmi les 55 chromosomes, 7 étaient courbés, 5 étaient très courbés avec un important angle d'inclinaison, pour ce cas, le centromère n'a pas été identifié avec les deux techniques.

Notons que d'autres études [20, 22] ont été réalisées pour la polarisation des chromosomes en se basant sur la localisation du centromère. La procédure de polarisation est nécessaire pour se conformer à la norme d'orientation des chromosomes adoptée par les cytogénéticiens (bras long « q » vers le bas); et à fournir une direction uniforme de lecture des caractéristiques pour le classificateur.

Le modèle de chaînes de Markov a été aussi employé pour la localisation du centromère. L'approche exposée dans [23] est basée sur les informations de bandes sans tenir compte des caractéristiques géométriques. Les caractéristiques relatives aux bandes chromosomiques sont aussi considérées comme uniques caractéristiques pour la classification dans d'autres travaux. Par exemple dans [24] une classification basée sur la résolution des bandes est adoptée par *Ravi et al.* Cette étude a montré que les aberrations numériques sont mieux détectées avec des bandes de faibles résolutions. Tandis que les bandes avec de hautes résolutions sont plus appropriées pour la détection des aberrations structurelles.

1.2.2 Extraction de l'axe médian du chromosome

Dans plusieurs travaux relatifs à l'identification du centromère et l'extraction d'autres caractéristiques, l'axe médian du chromosome défini comme l'ensemble des points qui séparent le chromosome en deux parties similaires a été d'abord estimé. Cette étape est nécessaire pour le calcul des caractéristiques géométriques (la longueur du chromosome et l'indice centromérique) et les informations relatives aux bandes chromosomiques (les densités du profil).

L'estimation de l'axe médian est basée principalement, sur la technique MAT proposée, initialement, par *Jing-fu et al.* [25]. La technique MAT a été appliquée dans plusieurs travaux [14, 19, 21, 26-28]. Cette technique est la plus utilisée dans l'extraction des caractéristiques géométriques du chromosome, vu qu'elle préserve les propriétés topologiques de ce dernier. Pour raffiner l'axe médian, des algorithmes tels que la squelettisation et l'algorithme d'amincissement sont appliqués. Le principe de l'algorithme d'amincissement est de supprimer suivant un ensemble de règles, des pixels appartenant à des objets noirs jusqu'à obtention d'une forme d'un pixel d'épaisseur.

Dans [14] deux techniques sont utilisées pour l'extraction de l'axe médian. La première technique est la MAT basée sur la squelettisation [29], et la seconde technique est l'application linéaire par morceaux (PWL : *Piece Wise Linear*), qui est selon les auteurs, plus appropriée pour les chromosomes courbés. Dans cette technique, les moments du second ordre des niveaux de gris de l'image du chromosome sont employés pour calculer l'orientation de l'axe principal. Une extension de l'axe obtenu est appliquée dans les deux approches pour avoir un axe médian continu. Les auteurs affirment que l'axe médian extrait en utilisant la technique MAT basée sur la squelettisation est plus fin comparé à celui obtenu avec la technique PWL. Les paramètres géométriques et les densités de profil sont considérés dans cette étude pour 5 classes de chromosomes (2, 4, 13, 19 et X). Pour un apprentissage performant avec un minimum de caractéristiques et dans le but d'éviter les redondances; les auteurs ont procédé d'abord à la sélection des caractéristiques les plus significatives en employant l'algorithme *Knock-out*. Ils ont rapporté un taux de classification supérieur à 98% pour la première technique (MAT) et un taux d'environ 97% en utilisant les paramètres extraits avec la seconde technique (PWL).

Dans [30] *Eskiizmirli et al.* ont exploré les ondelettes pour l'extraction des caractéristiques. La quantification des densités de profil a été faite par l'application de la technique de transformée discrète en ondelettes (DWT : *Discrete Wavelet Transform*). Les ondelettes ont été aussi utilisées dans [31], la méthode proposée par *Theerapattanakul et al.* emploie la technique dite contours actifs et les ondelettes multidirectionnelles (*B-spline*) dérivant des filtres isotropes de Laplace. Le contour du chromosome est extrait en premier lieu en utilisant les contours actifs. Cette technique est utilisée aussi pour l'extraction de l'axe médian du chromosome et le calcul des densités de profil. La technique *B-spline* est ensuite appliquée pour déterminer la courbure sur le contour, afin d'extraire le centromère et les deux extrémités du chromosome.

Dans une autre étude [32], les auteurs ont proposé une méthode basée sur l'ajustement de courbe (*curve fitting*) pour l'identification du centromère et le calcul de la longueur du chromosome. Le tableau 1.1 résume un ensemble d'études concernant les méthodes d'extraction des caractéristiques. Principalement les méthodes relatives à l'identification du centromère qui est un paramètre largement utilisé pour la classification.

Tableau 1.1 Résumé de quelques études concernant les méthodes d'extraction des caractéristiques

Références	Méthodes	Caractéristiques	Base de données # chromosomes	Taux de réussite %
<i>Madian et al.</i> [19]	MAT et PV	Centromère	55	78% (MAT) 90% (PV)
<i>Wang et al.</i> [20]	MAT	Centromère	2287	91.4% (identification du centromère)
<i>Moradi et al.</i> [15]	PV	Centromère	87	96%
<i>Moradi et al.</i> [21]	localisation automatique des points clés dans le squelette du chromosome	Centromère et longueur	219 (groupe E)	Non spécifié
<i>Theerapattanakul et al.</i> [31]	MAT <i>active contour</i> et la technique B-spline	Centromère longueur et densité de profil	Non spécifiée	Non spécifié
<i>Lerner et al.</i> [14]	MAT PWL	Caractéristiques géométriques et densité de profil	622, 5 classes (2, 4, 13, 19 et X).	98% pour MAT 97 % pour PWL
<i>Gregor et al.</i> [23]	Réseau de Markov	Centromère	6985 (179 cellules)	94% estimation correcte du centromère

1.3 Méthodes de classification des chromosomes basées sur les réseaux de neurones artificiels (ANNs)

Les modèles de classification des chromosomes basés sur les approches réseaux de neurones artificiels (ANNs) sont très adoptés pour l'automatisation de la routine d'établissement du caryotype et la détection des anomalies chromosomiques [9, 33, 34]. Les ANNs ont été employés seuls ou combinés à d'autres techniques, tel que la logique floue, les ondelettes et l'algorithme génétique.

1.3.1 Modèles basés sur le réseau perceptron multicouches

Plusieurs types d'ANNs ont été adoptés pour la classification des chromosomes; en particulier le perceptron multicouches (MLP) [6, 14, 15, 21]. Par exemple ce modèle d'ANN a été utilisé dans l'étude exposée dans [14]. Dans ce travail 5 classes de chromosomes (2, 4, 13, 19 et X) extraites de 150 cellules (qui incluent 622 chromosomes) du liquide amniotique ont été considérées. Après estimation de l'axe médian des chromosomes avec la technique MAT, 66 caractéristiques ont été extraites (64 DPs, L et IC). Plusieurs combinaisons de ces caractéristiques ont été employées pour tester le classificateur. Selon les résultats rapportés dans cette étude, les caractéristiques géométriques (IC, L) sont plus appropriées pour la classification comparées à celles relatives aux bandes (DPs). Un taux de classification de 98% a été rapporté.

La combinaison de MLPs a été adoptée dans d'autres travaux; nous citons à titre d'exemple, [35] (*Wang et al.*), [36] (*Graham et al.*), [37] (*Phil et al.*). Le travail discuté dans [35] combine l'approche ANN à l'algorithme génétique. *Wang et al.* ont développé un classificateur multi-étage qui inclue 8 ANNs de type multicouches. Ils ont appliqué un algorithme génétique pour définir la topologie du ANN y compris le nombre de couches cachées et le nombre de neurones d'entrées qui représentent les caractéristiques optimales. Les auteurs ont rapporté un taux de classification de 86.7%

1.3.2 Autres modèles d'ANNs pour la classification des chromosomes

D'autres types d'ANNs, comme le réseau de neurones probabiliste ont fait l'objet d'autres études pour l'analyse des chromosomes [38] (*Sweeney et al.*), [39] (*S.*

Rungruangbaiyokand et al.) et [7] (Poletti *et al.*). Un ANN probabiliste a été employé dans [39], la classification a été élaborée en deux étapes. Dans la première étape, les chromosomes sont classés en six groupes et dans la deuxième étape les chromosomes des six groupes de la première étape sont classés en 24 classes. Les caractéristiques chromosomiques utilisées dans cette étude sont la surface du chromosome, la longueur du chromosome, la surface des bandes, la densité de profil et la valeur maximale de la matrice des valeurs singulières (*maximum value of singular value matrix*).

Ces caractéristiques sont extraites de 60 images qui incluent 2760 chromosomes, le taux de classification était de 68.18% pour les cellules femelles et de 61.30% pour les cellules mâles. Les auteurs ont conclu que le réseau probabiliste permet un apprentissage rapide comparé au réseau à rétro-propagation. Le temps d'apprentissage rapporté pour le réseau probabiliste était de 3.50s.

Une approche qui combine les ondelettes au réseau de neurones a été proposée dans [40]. Dans ce travail, *Oskouei et al.* ont développé un classificateur, appelé *Wavelet Neural Network* (WNN) pour la classification des chromosomes du groupe E (16, 17, 18). Le vecteur des caractéristiques inclut la longueur du chromosome, l'indice centromérique et des caractéristiques extraites de la densité de profil. La technique MAT a été utilisée pour l'extraction des longueurs et le calcul des densités de profil, la base de données considérée dans cette étude est constituée de 450 chromosomes (150 pour chaque classe du groupe E).

Une comparaison des performances du WNN à celles d'un ANN sans les ondelettes a été faite dans ce travail. D'après les auteurs, le WNN améliore le processus d'apprentissage ainsi que les résultats de classification. Ils ont rapporté des taux de classification de 94% et 89% pour le WNN et le ANN, respectivement. Le WNN donne de meilleurs résultats, vu la caractéristique non linéaire du WNN qui concorde avec la non linéarité de la structure du chromosome.

Dans [30], une approche hybride qui combine deux réseaux de neurones, l'un à apprentissage supervisé et l'autre non supervisé, a été exposée. Le premier réseau est de type MLP avec deux couches cachées où un algorithme de rétro-propagation du gradient est employé pour l'apprentissage. Le second réseau est de type Kohonen avec 24 ANNs représentant les classes des chromosomes. Le MLP détecte les anomalies numériques et le second réseau détecte les anomalies de structure. *Eskiizmirliler et al.* ont préalablement, développé deux algorithmes, à savoir l'algorithme de squelettisation (OSA : *Object Skeletonizing Algorithm*) pour l'extraction des

caractéristiques géométriques et l'algorithme de détection d'objet (AODA : *Automatic Object Detection Algorithm*) pour la détection automatique des chromosomes. La transformée discrète en ondelettes (DTW) a été utilisée pour extraire les densités de profil. 178 images métaphasiques qui incluent 8818 chromosomes ont été utilisées dans cette étude. Les auteurs ont comparé les performances de leur classificateur hybride avec un autre classificateur de type *Rule Based Classifier* (RBC). Ce dernier consiste à enregistrer les pratiques et règles employées par les cytogénéticiens dans l'établissement du caryotype manuel, et d'imiter ces règles dans un classificateur automatique basé sur ces dernières, d'où l'appellation *rule based classifier*. Les auteurs ont rapporté des valeurs d'erreurs maximales de 33.1% et 36.7% pour le classificateur hybride et le RBC, respectivement.

Un classificateur de chromosomes qui combine le réseau de Hopfield et la logique floue a fait l'objet du travail exposé dans [41]. Le classificateur nommé CFHN (*Classifier Fuzzy Hopfield neural Network*) développé par *Ruan et al.* est basé sur l'approche neuroflou, qui combine l'approche réseau de neurone et la logique floue. Les auteurs ont trouvé que pour une base de données qui inclue 100 chromosomes pour chacune des 24 classes; le CFHN donne un taux d'erreur d'identification de zéro et un taux de 3.33% pour les cas non identifiés. Les caractéristiques considérées dans cette étude sont la longueur du chromosome, l'indice centromérique et la densité de profil.

Dans [27] un classificateur à deux étages pour la reconnaissance et la classification des chromosomes a été exposé, le classificateur inclue un multicouches et un Bayésien. Dans le premier étage les chromosomes sont classés en 9 groupes, en utilisant les caractéristiques géométriques (L et IC) extraites après estimation de l'axe médian en utilisant l'algorithme du point milieu (*midpoint algorithm*). Dans le second étage, les chromosomes de chacun des 9 groupes du premier étage sont classés par le classificateur Bayésien en utilisant les caractéristiques de bandes. Le taux de classification rapporté est de 85.6% pour le multicouches (le premier étage) et de 72.3% pour le Bayésien (le second étage). Les auteurs ont comparé les performances de leur classificateur hybride avec un classificateur Bayésien employé seul (qui donne un taux de classification de 60.7%); ils affirment que leur classificateur multi-étage donne de meilleurs résultats.

Une étude comparative a été présentée dans [42] pour trois classificateurs d'images de chromosomes en G-bande, à savoir un ANN multicouches, un

classificateur à base de la logique floue (*fuzzy rule based*) et un classificateur de type *template matching*. Les caractéristiques géométriques et celles relatives aux informations de bandes sont utilisées dans cette étude. Des taux de classification de 94.76%, 93.54% et 96.98% ont été rapportés par *Badawi et al.* pour le ANN multicouches, le classificateur à base de la logique floue et le classificateur *template matching*, respectivement.

Un système modulaire pour la classification automatique des chromosomes a été proposé dans [43]. Le système inclue un module d'extraction des caractéristiques à partir de l'axe médian, qui a été estimé par un algorithme nommé *modified vessel-tracking* [43], préalablement développé par les mêmes auteurs. Les caractéristiques extraites sont la longueur du chromosome, 64 DPs, 64 contours, la surface du chromosome et son périmètre. Notons que les deux dernières caractéristiques ne sont pas extraites de l'axe médian. Les auteurs ont évalué leur système avec une base de données de 119 images *Q-band* qui incluent 5474 chromosomes, cette base de données est téléchargeable gratuitement via ce lien [44]. La polarité des chromosomes a été déterminée avant la classification à base d'un multicouches, qui donne un taux de classification de 94%.

Dans [45] *Ravi et al.* ont proposé la classification d'images de chromosomes pour une sélection optimale des cellules de test (*metaphase spreads*). Un classificateur de type *RBC* a été employé. Trois caractéristiques ont été définies, à savoir, le rapport de la largeur, le rapport de la hauteur et le rapport de surface. Les chromosomes sont classés en quatre classes : la première classe désigne les chromosomes individuels droits; la seconde classe pour les chromosomes individuels courbés et les classes 3 et 4, désignent les chromosomes se chevauchant et les artefacts (objets non chromosomiques dans l'image, comme par exemple les taches de coloration), respectivement. Les taux de classification étaient d'environ 99,42%, 90,67%, 89,44% et 93,25% pour la première classe, la seconde classe, la troisième classe et la quatrième classe, respectivement.

Des ANNs additionnels ont été considérés pour la classification automatique des chromosomes, tel que le *Single-layer* [46] (*A. Narayanana et al.*). Le tableau 1.2 récapitule quelques études considérées dans ce chapitre, concernant les systèmes de classification des chromosomes basés sur les ANNs; en terme de méthodes (type d'ANN considéré), des bases de données utilisées et des résultats rapportés.

Tableau 1.2 Résumé de quelques études concernant les systèmes de classification des chromosomes basés sur les ANNs

Références	Méthodes	Base de données	Taux de classification (%)
Eskiizmirli <i>et al.</i> [30] (1996)	ANN Hybride Kohonen et MLP	178 images métaphasiques 8818 chromosomes	Taux d'erreur maximum 33.1%
Lerner <i>et al.</i> [14] (1995)	MLP	150 cellules avec 622 chromosomes	Taux de classification 98%
Graham <i>et al.</i> [36](1992)	Combinaison de MLP	8106 (base de données Copenhagen) 4569 (base de données Edinburgh) 5817 (base de données Philadelphia)	Taux d'erreur MLP Combinaison de MLP 6.9% 6.2% 19.3% 17.1% 25.6 % 23%
Wang <i>et al.</i> [35] (2009)	Classificateur à deux couches (une combinaison de 8 ANNs)	6900 chromosomes	86.7%
Ravi <i>et al.</i> [45] (2013)	Un classificateur (RBC)	192 images (7817 objets segmentés)	99,42% Pour la première classe 90,67% pour la deuxième classe 89,44% pour la classe 3 93,25% pour la classe 4
Ravi <i>et al.</i> [24] (2013)	RBC	68 images (822 chromosomes droits 1012 chromosomes (artefacts inclus)	73.08% pour les bandes à faible résolution 95.24% pour les bandes à haute résolution
Poletti <i>et al.</i> [43] (2012)	ANN à deux couches	119 images (Q-band) 5474 chromosomes	94%
Delie <i>et al.</i> [27] (2010)	Un classificateur multi-étage (un Multicouche et un Bayésien)	Non spécifiée	Classificateur Multi-étage 85.6% pour le Multicouche (1 ^{er} étage) 72.3% pour le Bayésien (2 ^{ème} étage) 60,7% Bayésien seul
S. Rungruangbaiyokand <i>et al.</i> [39](2010)	Un ANN Probabiliste	60 images 2760 chromosomes	68.18% pour des images de cellules femelles 61.30 % images de cellules mâles
Oskouei <i>et al.</i> [40] (2010)	WNN	450 chromosomes	94%
Ruan <i>et al.</i> [41] (2000)	CFHN	100 chromosomes pour chacune des 24 classes de chromosomes	un taux d'erreur d'identification de 0% un taux de cas non identifiés de 3.33%
Badawi <i>et al.</i> [42] (2003)	1- MLP 2- classificateur à base de la logique floue 3-Template matching	images G-band	94.76% pour le MLP 93.54% pour le classificateur à base de la logique floue 96.98% pour le classificateur template matching

1.4 Classificateurs à base d'autres algorithmes

Bien que les classificateurs des chromosomes basés sur les ANNs soient les plus répandus dans la littérature, d'autres approches ont été explorées et appliquées pour la classification des chromosomes. Nous exposerons dans cette section quelques classificateurs de chromosomes basés sur d'autres algorithmes, tels que les classificateurs basés sur l'approche SVM, la logique floue et la théorie Bayésienne. Nous citerons également un certain nombre de travaux concernant la classification des chromosomes marqués avec différentes techniques autre que le *Giemsa*, comme par exemple la technique M-FISH (*Multiplex-Fluorescence In-Situ Hybridization*).

1.4.1 Classificateurs à base de l'algorithme SVM

L'approche SVM ou séparateur à vaste marge est une approche qui découle des travaux de *Vapnik* [47]. Cette approche est basée sur la théorie d'apprentissage statique, elle a été utilisée pour la classification des chromosomes dans plusieurs travaux [5, 16-18]. Ce type de classificateur est nouveau comparé aux classificateurs basés sur les réseaux de neurones et le classificateur Bayésien. L'étude rapportée dans [16] expose une approche basée sur le SVM pour l'établissement du caryotype pour des images métaphasiques de type CGH (*Comparative Genomic Hybridization*). Selon cette étude le classificateur SVM présente de bonnes capacités de généralisation et requière un nombre limité d'échantillons pour l'apprentissage. Le classificateur en question donne un taux de classification de 90%.

Une étude comparative entre quatre méthodes de classification, à savoir le SVM, le ANN (un perceptron à deux couches), un classificateur Bayésien et un classificateur Bayésien naïf (NBC: *Naive Bayesian classifier*) [48] a été exposée dans [5]. Les auteurs ont appliqué ces quatre méthodes aux signaux FISH extraits à partir de 400 images FISH. Les taux de classification qui ont été rapportés sont : 98.1%, 98.8%, 98.4%, 94.0% pour le ANN, le Bayésien, le SVM et le NBC, respectivement. Les taux de classification pour le ANN et le SVM sont comparables, selon les résultats de cette étude.

La méthodologie présentée dans [17] est basée sur un classificateur de type SVM à deux étages; dans le premier étage, un classificateur SVM *contexte-indépendant* est utilisé pour chaque classe de chromosome. Les résultats des

classificateurs du premier étage sont employés comme entrées pour le deuxième étage qui représente une classification *contexte-aware* pour une post-classification. Un taux d'erreur de 6.68% a été rapporté pour 99 images *Giemsa*.

1.4.2 Classificateurs additionnels

La logique floue est également adoptée dans la classification des chromosomes [49, 50]. Dans [49] un classificateur non supervisé basé sur la logique floue pour des images M-FISH est exposé. Les auteurs ont comparé le taux de classification obtenu avec leur classificateur et celui d'un classificateur de type *Maximum Likelihood* (ML) [51]. Des taux d'environ 88% et 89.8% sont rapportés pour le classificateur à base de la logique floue et le ML, respectivement. Le ML est également utilisé dans [52], dans cette étude, *Schwartzkopf* et ses co-auteurs ont proposé un système d'identification des chromosomes basé sur une combinaison de la segmentation et la classification des chromosomes.

Dans les études rapportées dans [53, 54], des images de chromosomes de type multicolores sont employées pour la classification. Un classificateur Bayésien a été adopté dans [53] pour la classification des images de type M-FISH; ce classificateur donne un taux de classification de 91.86 %.

1.5 Discussion et conclusions

Selon les travaux exposés dans ce chapitre, les caractéristiques des chromosomes sont extraites, généralement, de l'axe médian. Par conséquent, la première étape pour extraire ces caractéristiques est l'estimation de l'axe médian du chromosome. La technique la plus utilisée pour l'extraction de cet axe est la MAT. Cette technique est largement répandue car elle préserve les propriétés topologiques du chromosome. L'estimation de l'axe médian est en particulier utilisée pour l'identification du centromère, le calcul de la longueur du chromosome (calculée dans la majorité des études comme la somme des pixels constituant l'axe médian) et l'extraction des informations de bandes représentées par les densités de profil.

Il est à noter qu'outre la technique MAT, qui était le sujet de plusieurs travaux, d'autres techniques ont été appliquées pour l'identification du centromère, telle que la technique PV. Cette technique donne un meilleur taux d'identification de la position du centromère comparée à la technique MAT, principalement pour les chromosomes courbés. Selon [19] un taux d'identification d'environ 90% pour la PV contre 78% pour la MAT a été observé.

Dans une autre étude [15] les auteurs ont rapporté un taux d'identification d'environ 96% quand la technique PV est appliquée. Des techniques additionnelles ont été considérées dans d'autres études, comme l'application linéaire par morceaux (PWL) et les chaînes de Markov. Une exactitude de 97% est obtenue avec la PWL et un taux de 94% pour l'estimation correcte du centromère a été rapporté dans [23] pour la méthode basée sur le modèle de Markov.

D'après les travaux passés en revue dans ce chapitre concernant l'automatisation du processus de classification des chromosomes à base des ANNs. Le MLP a été le plus adopté et donne un taux de classification maximum, comme montrer dans le tableau 1.2. Les taux de classification obtenus avec ce type d'ANN varient de 78% à 98%. Toutefois une multiplicité d'ANNs est considérée dans d'autres études. Les taux de classification rapportés pour les autres ANNs varient de 61.30% à 96.98%. On cite à titre d'exemple, un taux de 68.18% pour le réseau probabiliste et environ 94% pour le réseau de neurones combiné aux ondelettes (WNN).

Outres les classificateurs de chromosomes à base des réseaux de neurones, d'autres types de classificateurs ont été employés, tels que le SVM, le Bayésien et ceux basés sur la logique floue. Il est à noter que ces classificateurs sont, particulièrement employés pour les images chromosomiques multicolores, tels que les images M-FISH. Le classificateur SVM est nouveau comparé au classificateur ANN. Néanmoins, la comparaison exposée dans [5] montre que le SVM et le ANN donnent des taux de classification comparables.

Les résultats de la classification dépendent, essentiellement de la base de données employée. En effet, l'extraction des caractéristiques optimales pour la classification dépend de la qualité des images de chromosomes traitées. C'est pourquoi

il est assez difficile de comparer les performances de la classification des différentes méthodes.

Notre contribution par rapport aux systèmes existants, qui sont basés sur des implémentations softwares, est la réalisation d'un système sur puce (SoC) pour la classification des chromosomes qui exploite le parallélisme des réseaux de neurones. Dans ce contexte, nous présenterons dans le chapitre suivant une synthèse des différentes approches et flots de conception SoC.

2

Les systèmes sur puce

2.1 Introduction et définition

Avec l'évolution des technologies d'intégration, il est possible d'intégrer plusieurs centaines de millions de transistors sur une seule pièce de silicium. La complexité croissante des systèmes est une conséquence directe de cette évolution des technologies cibles vers des niveaux d'intégration plus élevés. Cette augmentation de taux d'intégration a permis de concevoir des circuits intégrés (IC : *Integrated Circuit*) qui englobent une multitude de fonctionnalités sur une seule puce; ces circuits intégrés sont définis comme systèmes sur une seule puce ou systèmes mono-puce, communément SoC (*System on Chip*) [55, 56].

L'approche mono-puce représente une alternative de conception de circuit très répandue, vu l'opportunité que cette approche offre aux concepteurs pour intégrer à la fois les parties softwares (SW) et hardwares (HW) sur une seule pièce de silicium. Ce qui réduit la consommation de puissance et la taille du circuit, pour des ICs de plus en plus miniaturisés comparés au système sur carte. L'approche SoC permet la réduction du temps de conception et donc le temps de mise sur le marché (*TTM : Time to Market*).

Un des exemples de SoC est le système embarqué qui est un système autonome spécialisé, conçu pour une tâche bien précise. Un système embarqué doit répondre à certaines contraintes de temps réel. Ces systèmes sont de plus en plus adoptés dans divers domaines, entre autre : les multimédias, les transports, la défense, l'aéronautique, la santé, l'énergie et la télécommunication.

2.2 Architecture des SoCs

L'architecture d'un SoC, représentée par la figure 2.1, est composée d'un ensemble d'éléments assurant diverses opérations, une logique d'interconnexion est ajoutée pour assurer l'interfaçage de ces composants. Sur le même circuit on retrouve généralement un ou plusieurs microprocesseurs, des mémoires, des coprocesseurs, des IPs (*Intellectual Properties*), des réseaux de communication et des périphériques d'entrée/sortie [55].

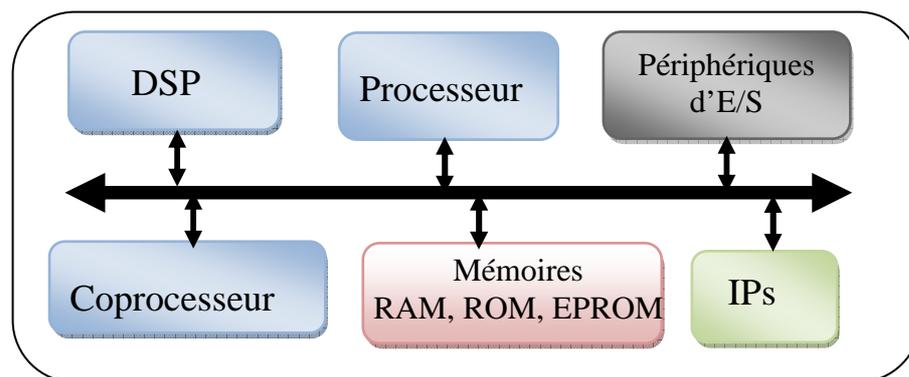


Figure 2.1 Architecture d'un système sur puce

2.3 Les méthodologies de conception SoC /Approche SoC

L'approche SoC est basée sur la translation de spécifications au niveau algorithmique d'une application en un système électronique ou circuit intégré (IC) réalisant la fonctionnalité désirée. Plusieurs méthodologies de conception ont été proposées pour le développement d'un flot de conception SoC complet, partant d'une spécification au niveau système de l'application et allant jusqu'au prototypage sur un support hardware

2.3.1 Flot de conception SoC

La figure 2.2 représente les différentes étapes du flot de conception SoC dans sa généralité. On représente ici deux aspects de la conception SoC, à savoir la conception générique où l'architecture générée ne contient que des parties hardware, et la conception Co-design où les deux parties HW et SW sont intégrées sur la même puce (c'est le principe des systèmes embarqués). Notons que les étapes pour les deux démarches sont identiques, excepté l'étape d'exploration d'architecture.

La première étape du flot de conception SoC est la description des spécifications du système. Ces dernières représentent ses fonctionnalités globales (description algorithmique). Dans cette étape l'ensemble des tâches envisagées sont décrites d'une manière générale indépendamment des ressources matérielles. Ces spécifications résument le comportement du système qui est exprimé par des tâches (processus) parallèles dans la phase suivante de la conception.

Après décomposition des fonctionnalités en sous-systèmes réalisant des tâches concurrentes, une exploration d'architecture est effectuée dans la troisième étape du flot de conception, le choix de l'architecture est effectué à ce niveau. C'est cette phase qui différencie la démarche générique de celle du Co-design. Pour la conception générique toutes les parties de l'architecture sont réalisées en hardware. Un partitionnement HW/SW est requis pour une conception Co-design ; il s'agit dans ce cas de faire le choix des tâches qui seront réalisées en HW et celles implémentées en SW. Le choix de la cible (support hardware) qui répond aux spécifications et au partitionnement établi se fait à ce niveau.

L'étape qui précède l'implémentation proprement dite est celle du raffinement de l'architecture; où chaque partie de l'architecture est modélisée soit en RTL (*Register Transfer Level*) pour les parties qui seront réalisées en HW, ou génération du code pour les parties de l'application envisageables en SW (logiciel ou code qui sera exécuté par le processeur). Pour les parties qui seront réalisées en HW, la modélisation RTL est faite en langage HDL (Hardware Description Language) tel que le langage VHDL ou Verilog, ces langages permettent de décrire le système avec un niveau d'abstraction élevé. Ils représentent des outils de base pour la conception de systèmes logiques intégrés implémentés sur des composants reconfigurables tels que les FPGAs ou des circuits intégrés spécifiques ASICs.

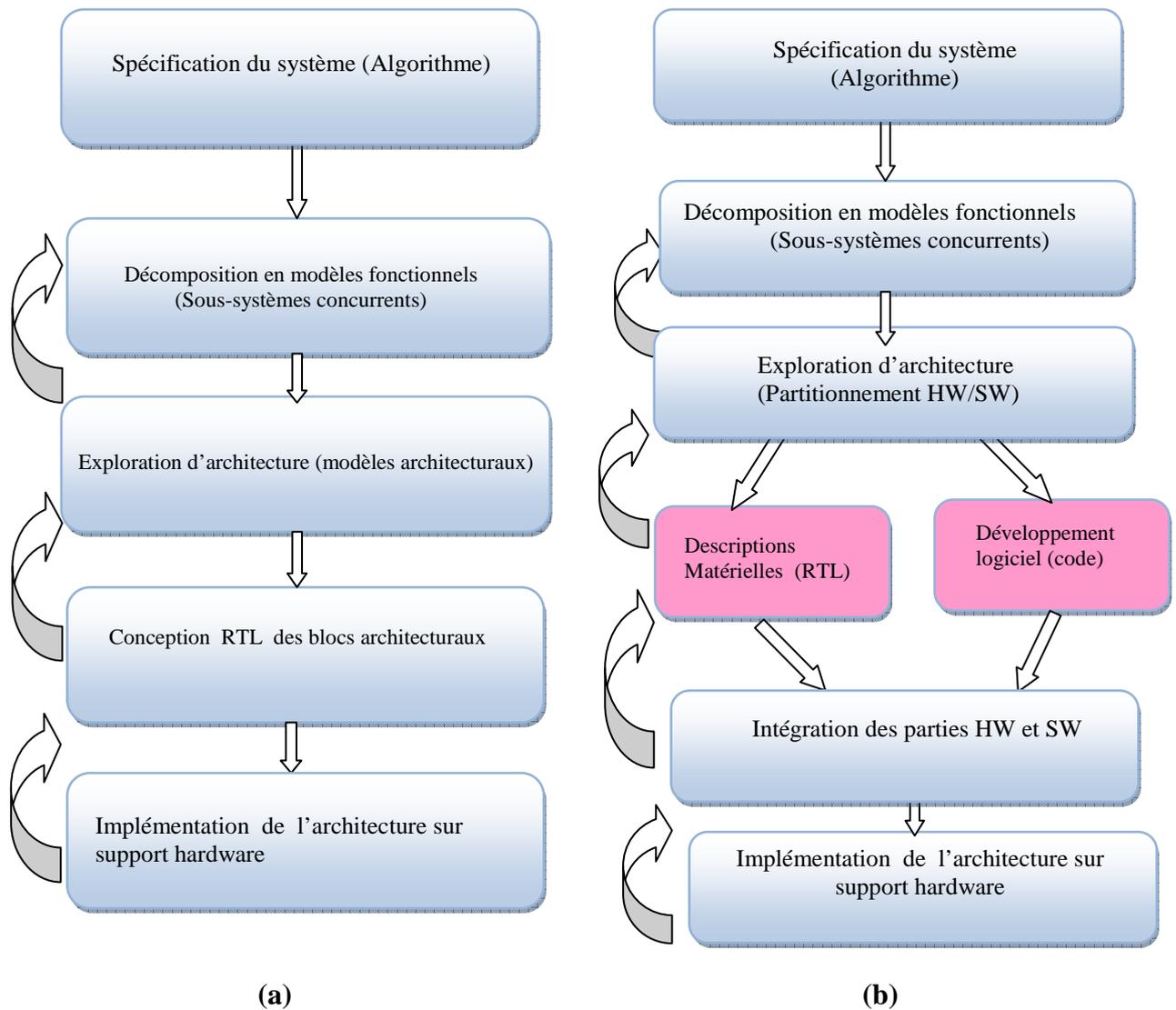


Figure 2.2 Flot de conception SoC
(a) Conception générique (b) Conception Co-design

La dernière étape est celle de l'implémentation sur la cible choisie, cette étape constitue en elle-même un autre flot de conception qui dépend du type de la cible (support hardware), qui peut être un FPGA, un CPLD (*Complex Programmable Logic Device*) ou un ASIC. Nous détaillerons dans la section suivante, les phases relatives à la dernière étape, en particulier pour les FPGAs (support de notre implémentation) et les ASICs, vu que ces deux supports hardware sont les plus utilisés dans la conception des SoCs. Une rétroaction est effectuée dans chaque étape pour répondre aux contraintes exigées dans la conception.

2.3.2 Approche réutilisation d'IPs

La complexité croissante de la conception des systèmes sur puce, en particulier pour la conception ASIC impose aux concepteurs de SoC l'exploration de solutions permettant la flexibilité et le prototypage rapide pour économiser le temps et l'effort. Cette section traite l'une de ces solutions, à savoir le concept de réutilisation d'IPs. Cette approche est basée sur la réutilisation de blocs architecturaux portables, décrits au niveau RTL, dont la description est indépendante de la technologie. L'architecture d'un SoC composé de ces IPs peut être implémentée sur un ASIC ou un FPGA avec un minimum de changements dans le code RTL lors du portage du FPGA vers ASIC.

Il est convenu dans la conception SoC que le choix entre une plateforme FPGA et un ASIC, comme support d'implémentation est imposé par l'application ciblée et les performances requises. Plusieurs critères sont pris en considération, entre autres la consommation de puissance, la surface, le coût et le *TTM*. Avant l'apparition des FPGAs [57] et leur déploiement sur le marché, la conception SoC était limitée aux grandes firmes qui réalisent un grand volume de production sur ASICs; pour plusieurs raisons, entre autres, les coûts élevés des outils de développement et le NRE (*Non Requiring Engineering*) élevé pour la conception ASIC.

La conception sur des plateformes FPGAs requière un coût NRE réduit comparé aux ASICs. Ces plateformes sont largement utilisées dans plusieurs applications où un système complet peut être mis en œuvre sur une seule puce. Toutefois, elles présentent un inconvénient majeur qui est une large consommation de puissance par rapport aux ASICs ou une conception *full custom*. D'autre part, dans une conception ASIC, les phases de vérification nécessitent autant de temps que celui requis pour la description, la synthèse et l'implémentation du design.

Néanmoins, une approche hybride qui combine les deux supports, où un FPGA est utilisé pour le prototypage du design qui sera ciblé pour un ASIC; peut être une solution rapide et facile pour la validation fonctionnelle. Ce concept est rendu possible avec la réutilisation d'IPs décrits au niveau RTL. En adoptant cette approche, les concepteurs sont capables de prototyper leurs systèmes sur plateforme FPGA pour la validation fonctionnelle, dans des conditions réelles, permettant la détection de défauts plus tôt dans le processus de conception et un gain de temps significatif.

2.3.3 Implémentation sur FPGA

2.3.3.1 Avantages et inconvénients des circuits FPGAs

La technologie FPGA, cible plusieurs applications du domaine public. Un support hardware de type FPGA peut être utilisé comme support d'un produit final pour une production de faible volume. Dans la conception de systèmes à base de FPGAs il n'y a pas de temps de fabrication comme pour les ASICs (pas de fabrication de masques par rapport aux ASICs), ainsi la conception peut être testée et évaluée rapidement, permettant un cycle de développement et un temps de conception plus court, en conséquence une réduction du coût de conception.

En ce qui concerne les outils de CAO (Conception Assistée par Ordinateur), les fournisseurs de circuits FPGAs offrent des outils qui aident les concepteurs à entreprendre les étapes du flot de conception. On cite à titre d'exemple, le logiciel ISE (*Integrated Software Environment*) de Xilinx [58] et QuartusII d'Altera [59]. Des versions d'évaluations de ces outils sont disponibles pour téléchargement gratuit, les versions complètes sont également disponibles pour des prix raisonnables comparés aux outils de conception ASIC. Toutefois, lorsque la conception est destinée pour des applications de faible puissance et un grand volume de production, les FPGAs ne sont pas appropriés, ce qui limite l'utilisation des FPGAs pour de faibles volumes de production par rapport aux ASICs.

2.3.3.2 Flot de conception FPGA

La figure 2.3 illustre le flot de conception FPGA à partir des descriptions RTL qui modélisent l'architecture à implémenter jusqu'à la configuration de la carte FPGA. Les outils de conception FPGA sont généralement des outils graphiques (*GUI : Graphical User Interface*) et ne sont pas basés sur des *scripts* comme c'est le cas pour les outils de conception ASIC.

Nous exposerons dans ce qui suit le flot de conception FPGA de Xilinx (figure 2.4 (a)), qui est utilisé dans notre conception. Ce flot inclut, principalement, la synthèse, le placement&routage et la programmation de la carte. Les descriptions qui représentent la fonctionnalité du design sont écrites en HDL (VHDL ou Verilog) ou en schématique et sont synthétisées par la suite. La synthèse est un processus automatisé

permettant de faire passer la représentation d'une architecture du domaine comportemental au domaine structurel. Le code RTL est converti en *netlist* structurelle, désignée et optimisée pour les composants de Xilinx en utilisant les bibliothèques de Xilinx. Ces bibliothèques contiennent des primitives, telles que les portes logiques, des flaps-flops, des Look-Up-Tables (*LUTs*) et des Macros (qui englobent plusieurs éléments, tels que les BRAMs, les multiplieurs, les blocs DSP (*Digital Signal Processor*)....). Chaque primitive a des caractéristiques spécifiques qui définissent son symbole, sa description et le nom de la bibliothèque.

Après synthèse, le design est mappé sur la cible FPGA, dans cette phase les éléments logiques (la *netlist*) sont associés avec les éléments physiques (*Slices*) des composants du FPGA. La description physique du design est placée et routée, par la suite. A l'issue de cette phase, un fichier (*NCD*) est généré, ce dernier sera transformé par l'outil BitGEN en *bitstream* pour la configuration du FPGA. Une vérification sur circuit peut être effectuée en envoyant le fichier binaire (*bitstream*) sur le FPGA via l'outil *Impact* du flot de Xilinx. La figure 2.4 (b) illustre les détails des fichiers générés dans chaque étape de ce flot de conception.

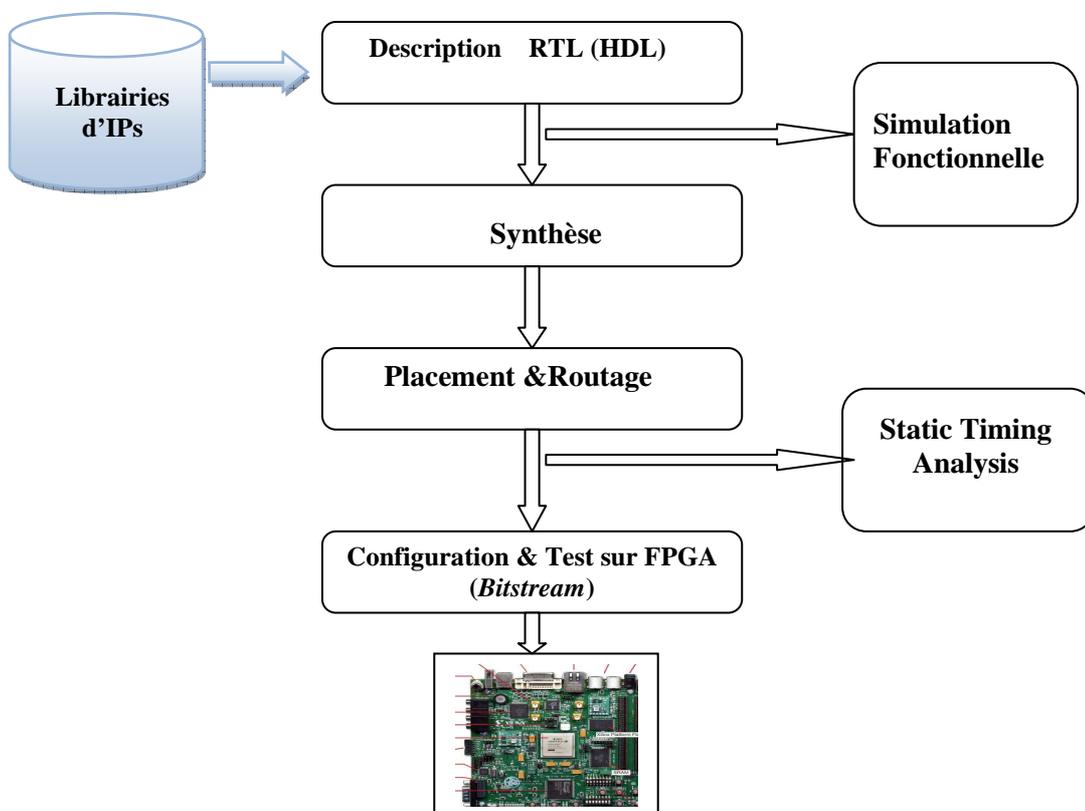


Figure 2.3 Flot de conception FPGA

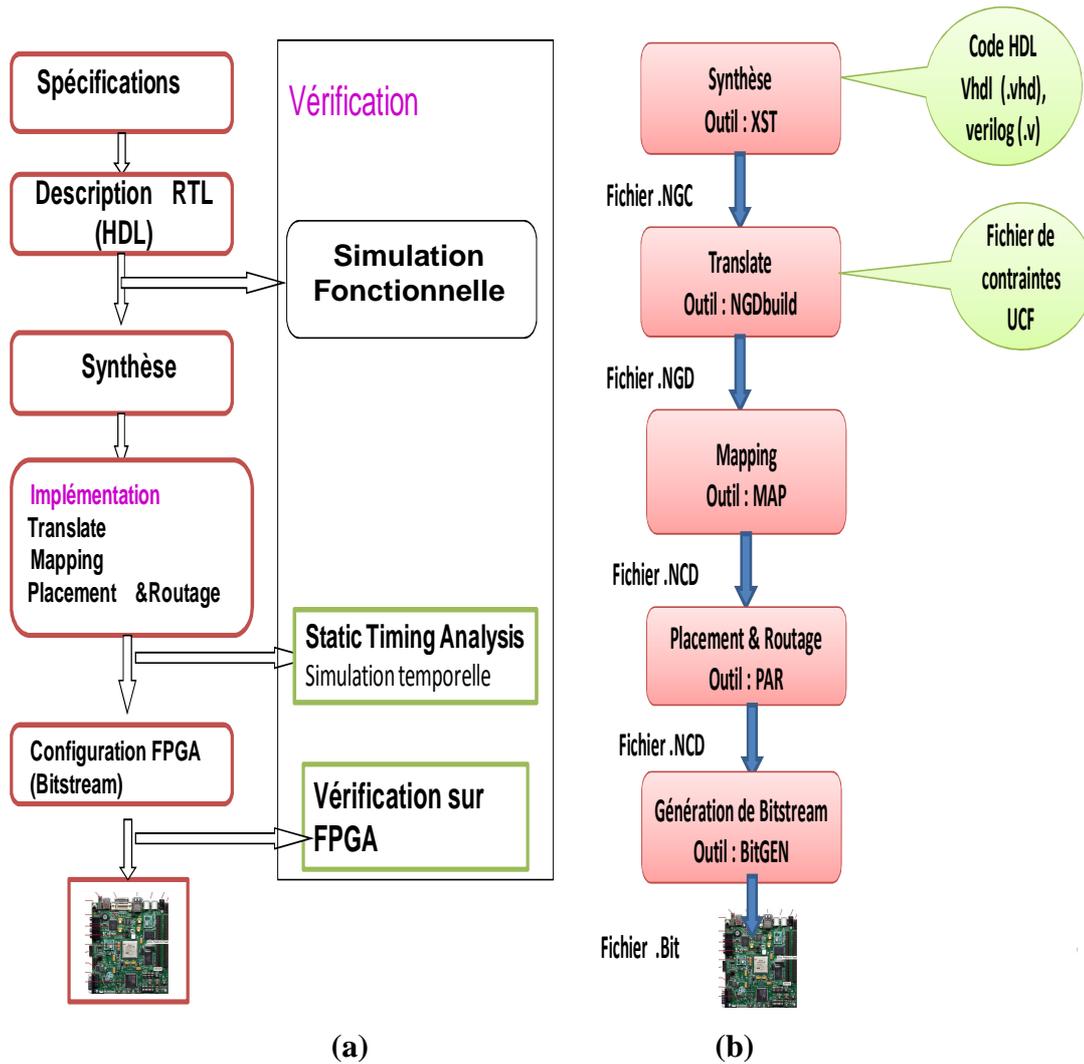


Figure 2.4 (a) Flot de conception FPGA de Xilinx (b) Détails des fichiers dans chaque processus

2.3.3.3 Les vérifications dans les FPGAs

Plusieurs niveaux de vérification sont considérés dans la validation d’une conception sur FPGA, à savoir :

La simulation fonctionnelle : Elle permet de tester la fonctionnalité du design et de corriger les erreurs de conception au niveau comportemental, on cite à titre d’exemple, les outils *ISE Simulator* de Xilinx et *Modelsim* de Mentor Graphics [60].

L’analyse temporelle statique (STA : *Static Timing Analysis*) : Après la phase de placement&routage, une analyse temporelle statique est appliquée pour

déterminer le chemin critique dans la conception. La STA permet de vérifier que la conception répond aux contraintes de temps indiquées dans le fichier UCF (*User constraints file*) avant la phase d'implémentation. L'outil de Xilinx utilisé pour l'analyse temporelle est *Timing Analyser* ou *TRACE*.

La simulation temporelle après placement&routage : Après placement&routage, le design est re-simulé avec une simulation temporelle qui tient compte des délais entre les portes logiques. Dans cette phase le fichier de placement &routage (*NCD*) est traduit en fichier interprétable par l'outil de simulation. La simulation temporelle détermine le chemin critique et vérifie les relations temporelles dans le design.

Vérification sur FPGA (test physique) après configuration : C'est la vérification du design dans des conditions de fonctionnement réelles. Elle est effectuée après programmation du circuit FPGA, en envoyant le fichier de configuration *bitstream (.bit)* via le câble de configuration, en utilisant l'outil *impact* de Xilinx.

2.3.4 Implémentation sur ASIC

2.3.4.1 Les ASICs

Les ASICs sont des circuits intégrés spécialisés, conçus pour des applications spécifiques avec des architectures personnalisées pour un domaine d'application bien ciblé. La production ASIC est réservée pour de grands volumes de production, vu le coût de fabrication élevé. Dans certains cas, un circuit spécifique est indispensable pour un petit volume de production avec un coût de production très élevé. Il existe trois types d'ASICs comme illustré sur la figure 2.5, à savoir le *full custom*, le *semi-custom ASIC (gate array et standard cells)* et le *Structured ASIC*.

La conception *full custom* est une approche dont le circuit intégré est conçu au niveau transistor en spécifiant le layout de chaque transistor intégré dans le circuit. Cette approche présente plus de performances (surface et consommation de puissance réduites). En parallèle elle est plus complexe et le temps de réalisation est relativement lent. Le coût de fabrication de ce type d'ASIC est élevé, ce qui limite son utilisation pour des productions de grands volumes.

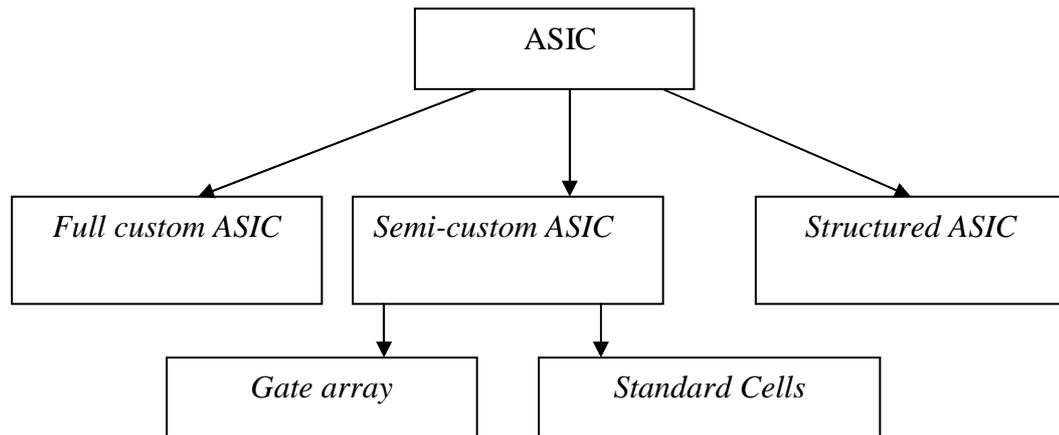


Figure 2.5 Types d'approches de conception ASIC

La conception *semi-custom (standard cell)* est basée sur les *standard cells* (portes logiques, flip-flop, adders, MUX etc.), qui sont des composants logiques conçus en *full custom* avec une technologie ciblée. Ces composants sont organisés dans une bibliothèque (PDK : *Process Design Kit*) qui sera utilisée comme entrée de l'outil de conception ASIC.

Le troisième type est le *structured ASIC* [61, 62] qui est une approche intermédiaire entre les FPGAs et les ASICs. Elle permet de réduire le *TTM* et le coût de conception. L'ASIC structuré a la structure d'un FPGA, c'est la programmation qui diffère, il est programmable par masque en configurant une ou plusieurs couches métalliques.

Notons que l'approche *standard cells* ASIC est l'approche la plus utilisée pour la conception ASIC, l'approche *Structured ASIC* reste modérément adoptée. Une comparaison détaillée des performances des différents types de conception ASIC est donnée dans [63].

Le tableau 2.1 représente une comparaison des performances en termes de complexité, densité d'intégration, vitesse, coût et consommation de puissance pour chaque technologie (la technologie FPGA incluse).

Tableau 2.1 : Comparaison des performances des différentes technologies

Technologie Performances	ASIC			FPGA
	Full custom	Standard Cell	Gate Array	
Densité d'intégration	Haute	Medium	Medium	Faible
Complexité	Très Haute	Haute	Medium	Faible
Consommation de puissance	Faible	Medium	Medium	Haute (static power)
Vitesse	Très haute	Haute	Medium	Faible
Coût	Très haute	Haute	Medium	Faible
Flexibilité	Faible	Faible	Medium	Haute

2.3.4.2 Avantages et inconvénients des ASICs

La conception ASIC présente certains avantages, à savoir :

- Faible coût pour une production de volume important.
- Faible consommation de puissance.
- Haute vitesse de fonctionnement.
- Optimisation du circuit réalisé, vu que le circuit est conçu pour une application bien spécifique.
- Réduction de la taille du circuit.
- Contrairement aux FPGAs, les ASICs permettent des conceptions analogiques et mixtes.
- Plusieurs techniques d'optimisation de puissance [64] sont applicables en ASIC, comme le *clock gating (CG)*, *multi-Vth*, *operand isolation*, *power gating*..... Excepté la technique *CG*, les autres techniques ne sont pas applicables en technologie FPGA.

Toutefois les ASICs présentent des inconvénients, nous citerons principalement :

- Un lent cycle de développement, et donc un *TTM* important.
- Les outils de conception ASIC (*front-end* et *back-end*), entre autres les outils de Cadence [65] et Synopsys [66] sont très coûteux.
- Les étapes de fabrication et de vérifications ASIC sont coûteuses.

2.3.4.3 Flot de conception ASIC

La figure 2.6 illustre le flot de conception ASIC; les différentes étapes incluses dans ce flot sont :

La synthèse logique : La synthèse logique est le processus de conversion du code RTL en *netlist*, basée sur les bibliothèques de *standard cells* (portes logiques, flip-flop, blocs RAM.. etc.) conçues dans une technologie donnée. Les outils de synthèse ASIC les plus utilisés sont *Design Compiler* (DC) [67] de Synopsys et *RTL Compiler* (RC) [68] de Cadence, la synthèse est basée sur des scripts, qui listent les différentes commandes.

Le Floorplan: Permet à un concepteur d'ajuster le placement de tous les composants dans un design et de changer le placement si nécessaire.

Le placement&Routage : Après synthèse, des fichiers sont générés pour être utilisés avec l'outil de placement&routage (comme l'outil *Encounter* de Cadence), par exemple, le fichier *netlist* en Verilog qui est une description structurelle du design au niveau *standard cells*. Les *standard cells* sont placées dans des locations appropriées et les connexions entre ses différentes cellules sont faites dans cette étape.

Analyse après synthèse : Les rapports de synthèse relatifs à *area*, *power*, *timing* et *gates* sont générés sous forme de fichiers, pour pouvoir analyser le design et vérifier que l'implémentation répond aux spécifications de *timing* et de surface requises.

Optimisation : Permet d'avoir une implémentation optimisée en surface et en consommation de puissance.

2.3.4.4 Les vérifications dans les ASICs

Nous donnerons, brièvement dans ce qui suit, les étapes de vérification après synthèse dans une conception ASIC :

- La simulation fonctionnelle : Permet de vérifier et tester la fonctionnalité du design.
- Analyse statique temporelle : Utilisée pour déterminer le chemin critique dans la conception et vérifier que la conception répond aux contraintes temporelles.
- La vérification formelle (*Equivalency checking*) : Vérifie si deux versions d'une conception sont fonctionnellement équivalentes. Par exemple, une conception RTL versus une conception post-synthèse.

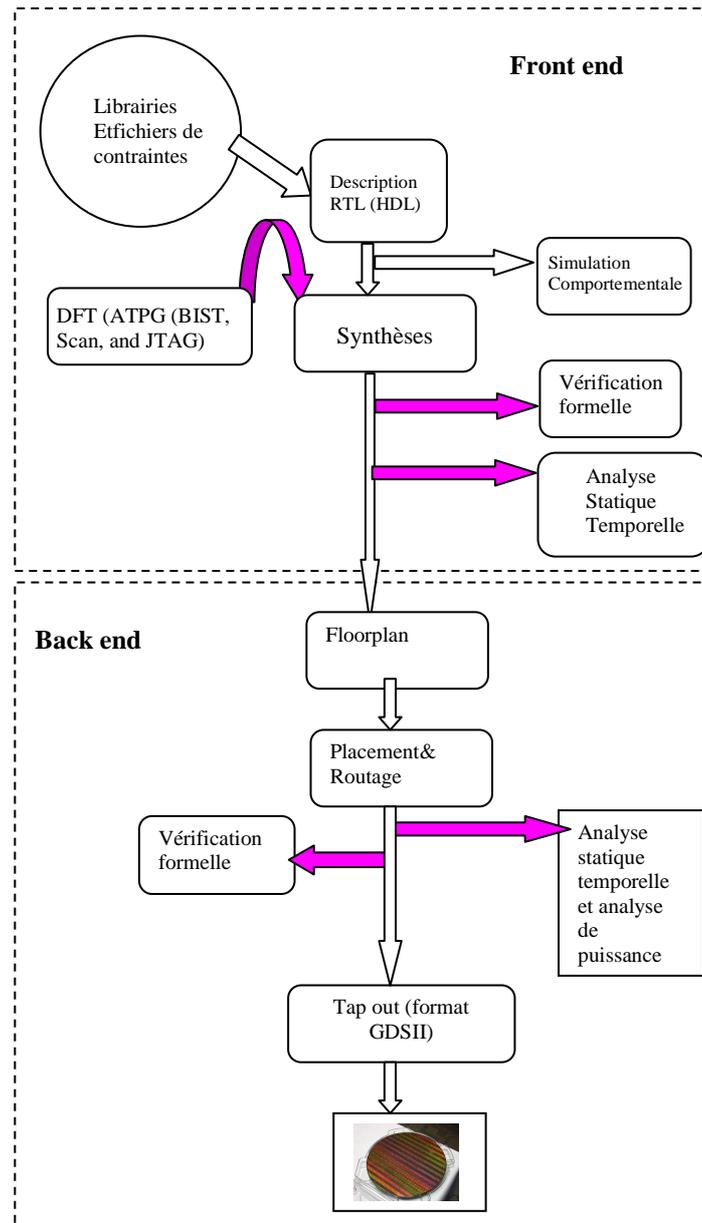


Figure 2.6 Flot de conception ASIC

- Insertion de la logique de test (*DFT : Design For Test*) : Plusieurs tests sont considérés, à savoir :

ATPG- Automatic Test Pattern Generation: Vecteurs de test générés pour tester une partie du circuit.

BIST- Built-In Self Test: Utilisé pour tester la fonctionnalité des ressources de mémoire (pour les RAM, spécifiquement).

Internal Scan Chain : Des registres à décalage (*shift register*) internes sont créés pour tester la fonctionnalité d'une partie du circuit.

2.4 La synthèse haut niveau

La complexité des applications ciblant les SoCs a contraint les concepteurs à chercher des solutions et méthodologies permettant le passage à des niveaux d'abstraction plus élevés. Aujourd'hui la conception est au niveau système; ce concept est rendu possible avec des outils d'automatisation de conception électronique de système (*EDA : Electronic Design Automation*), de type HLS (*High Level Synthesis*) [69, 70]. Ces outils permettent l'automatisation du passage d'une description haut niveau (système) vers des descriptions destinées pour des implémentations physiques; ciblant des supports hardware différents tels que les FPGAs ou les ASICs.

Le concept de la conception HLS est illustré sur la figure 2.7, il s'agit d'une transformation des spécifications fonctionnelles décrites au niveau algorithme en C, C++ ou SystemC (code C ou C++ avec des spécifications hardware) en descriptions RTL (VHDL ou Verilog). L'approche HLS transforme une macro-architecture en une microarchitecture avec un gain en temps de développement (conception du design) et un gain en temps de vérification, vu qu'une Co-simulation est effectuée en C et en RTL. Le même *testbench* utilisé pour la vérification du code C et utilisable pour la simulation de la description RTL générée automatiquement.

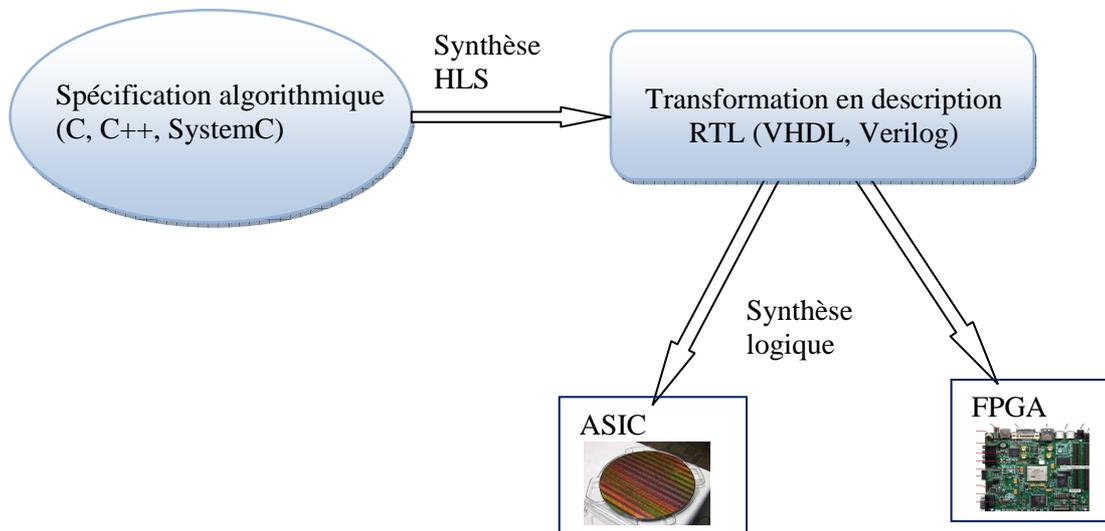


Figure 2.7 Approche HLS

Dans cette approche la spécification initiale n'est pas décomposée en sous-systèmes. Plusieurs outils commerciaux sont proposés par des fournisseurs d'outils d'aide à la conception, tel que Vivado HLS de Xilinx [71], Symphony HLS de

Synopsys [66], Stratus HLS de Cadence [65], et Catapult HLS de Mentor Graphics [61]. Les figures 2.8 et 2.9 illustrent des exemples de flots de conception de Xilinx Vivado HLS et de Stratus HLS de Cadence, respectivement.

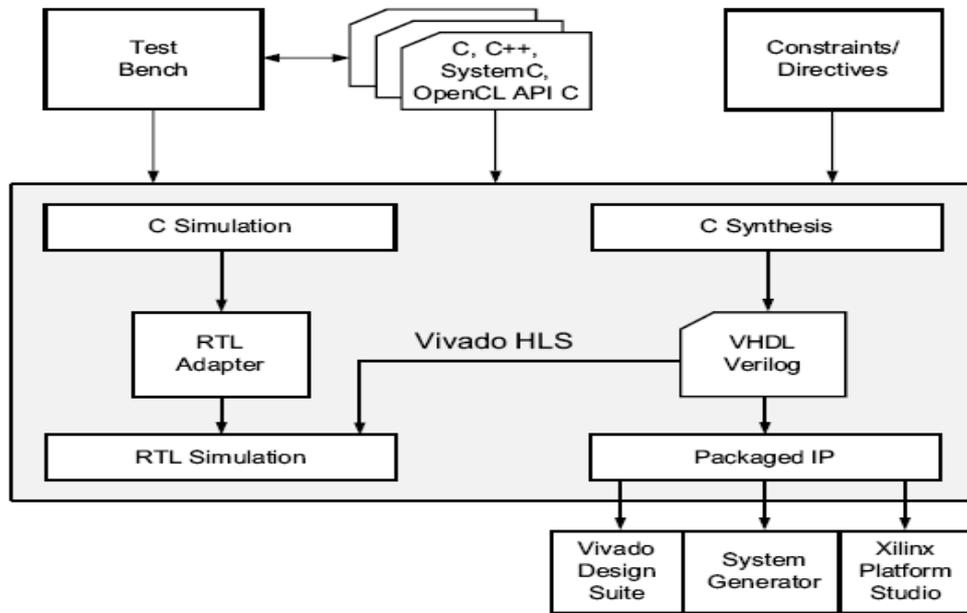


Figure 2.8 Flot de conception Vivado HLS [71]

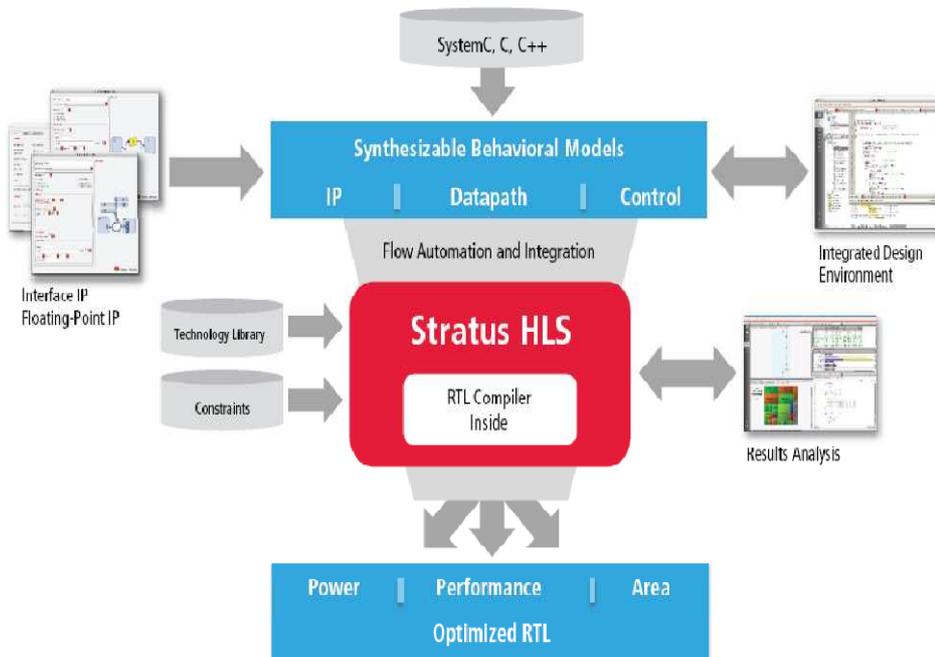


Figure 2.9 Flot de conception Stratus [65]

2.5 Conclusion

La conception des SoCs est complexe et nécessite l'exploration de nouvelles approches pour satisfaire les contraintes croissantes de conception et les exigences des nouvelles applications. Afin de remédier à la complexité de conception des SoCs, les développeurs de logiciels proposent des outils complémentaires de conception, permettant la transposition d'une description algorithmique ou description fonctionnelle de l'application en un système électronique implémenté sur support hardware.

Dans ce chapitre, nous avons passé en revue les différentes approches et flots de conception SoC, en particulier les conceptions ciblant les FPGAs ou les ASICs. Dans le prochain chapitre, nous exposerons l'approche méthodologique que nous avons suivie pour l'implémentation hardware d'un système automatique de caryotype qui repose essentiellement sur l'approche SoC.

3

Approche Méthodologique de développement du SoC-AKS

3.1 Introduction

Le présent chapitre décrit l'approche méthodologique proposée pour une implémentation hardware d'un système automatique de caryotype. Nous présenterons dans une première partie, une introduction au concept d'un système de classification des chromosomes humains et ses différentes parties. Dans la deuxième partie de ce chapitre, nous exposerons succinctement le diagramme en Y dérivant les trois vues d'une architecture. En effet, notre approche de conception découle de cette représentation. Nous détaillerons par la suite, les différentes étapes du flot de conception suivi pour l'implémentation hardware du AKS, en particulier les parties extraction des caractéristiques et classification. Chaque sous-système est modélisé par une architecture hardware dans le SoC-AKS.

3.2 Système AKS proposé

3.2.1 Concept d'un système de classification des chromosomes

Un système automatique de caryotype [72] est dédié pour une classification des chromosomes dans un format standard, permettant la détection d'éventuelles

anomalies, sans l'intervention d'un opérateur cytogénéticien. La figure 3.1 illustre le concept de base d'un système de classification des chromosomes qui comporte, principalement, trois étapes: le prétraitement des images métaphasiques, l'extraction des paramètres (caractéristiques) et la classification des chromosomes.

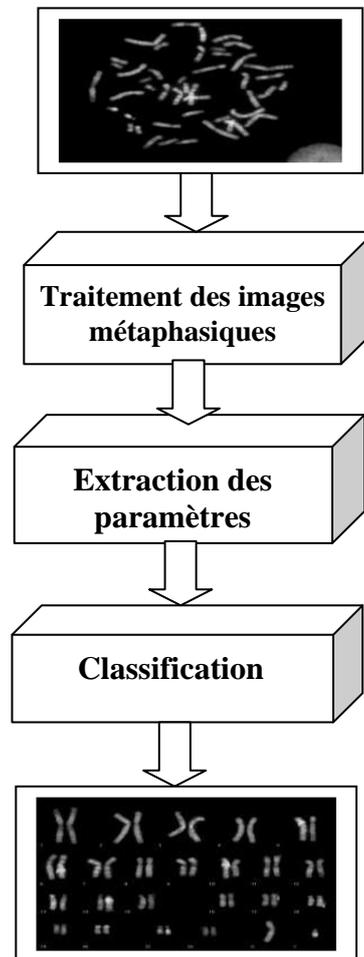


Figure 3.1 Les principales étapes d'un AKS

3.2.2 Description du AKS proposé

Dans ce qui suit nous décrivons brièvement les principales parties du système proposé pour l'analyse et la classification automatique des chromosomes, qui est basé sur le concept de tous systèmes AKS. La figure 3.2 illustre ses principales composantes; on distingue trois sous-systèmes: le sous-système traitement d'images, le sous-système extraction des paramètres et finalement le sous-système classification des chromosomes, ce dernier est basé dans notre cas sur l'approche ANN.

Après acquisition des images métaphasiques à partir d'un microscope USB, doté d'une caméra\capteur intégrée, ces dernières seront stockées dans une mémoire externe et traitées dans l'étape prétraitement, où la qualité de l'image est améliorée pour faciliter la segmentation des chromosomes. Après séparation des chromosomes du background de l'image, leurs squelettes seront élaborés pour être utilisés dans la phase d'extraction des paramètres. L'axe médian qui représente une approximation de la courbe centrale du chromosome est extrait, en premier lieu. Plusieurs techniques ont été proposées dans la littérature pour l'estimation de l'axe médian et l'extraction des paramètres chromosomiques. Dans notre système la squelettisation avec l'algorithme de Marthon est employée, ce dernier s'avère plus approprié que d'autres algorithmes de squelettisation selon l'étude exposée dans un travail réalisé précédemment dans notre laboratoire [11]. En effet, le squelette du chromosome avec l'algorithme de Marthon présente moins de bifurcations.

Trois caractéristiques sont extraites dans l'étape extraction des paramètres, à savoir la longueur du chromosome, la densité de profil et le centromère; l'identification du centromère permet de calculer l'indice centromérique. Ces paramètres sont les composantes du vecteur des caractéristiques qui sera utilisé dans la phase de classification, qui constitue l'étape principale du système. Dans cette étape chaque chromosome est assigné à sa classe correspondante afin d'établir le caryotype.

Divers approches et modèles ont été utilisés pour la classification des chromosomes, comme rapporté dans le chapitre 1. Pour notre système la partie classification est basée sur l'approche ANN, le réseau adopté étant une combinaison de deux ANNs de type Kohonen (comme le montre la figure 3.2).

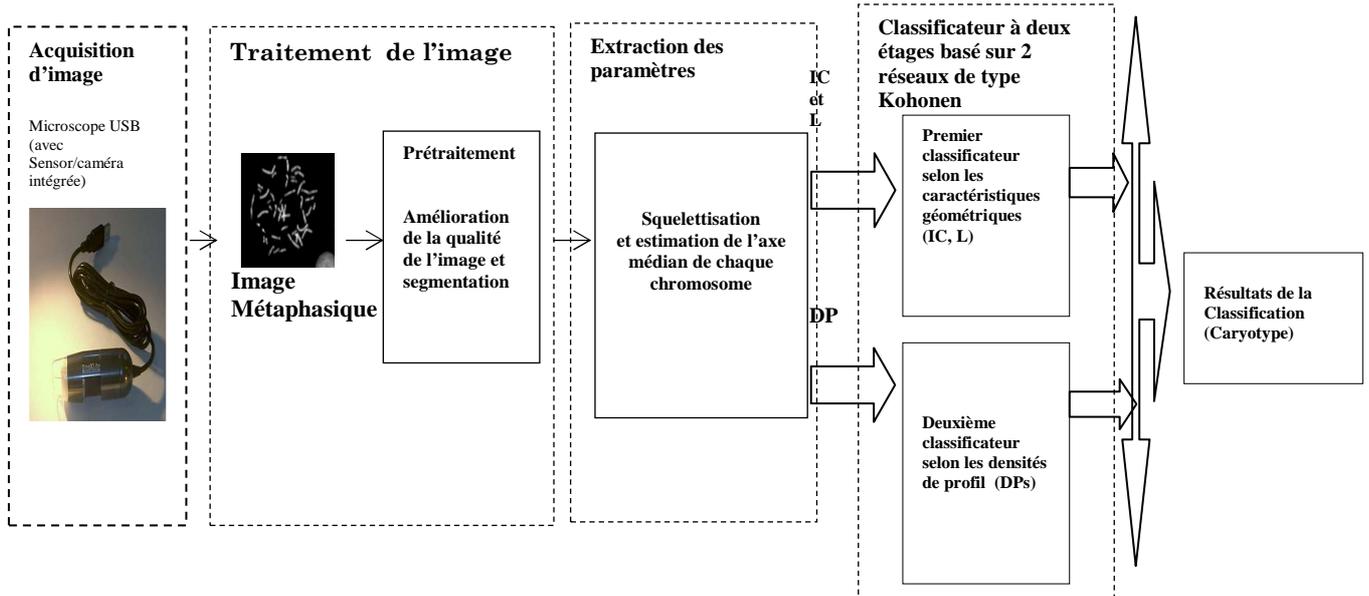


Figure 3.2 Schéma synoptique du système proposé pour la classification des chromosomes

3.3 Approche méthodologique de développement

L'approche méthodologique que nous proposons permet l'élaboration d'un système de classification des chromosomes humains sur un support hardware de type FPGA. On s'intéresse plus particulièrement à l'approche méthodologique qui aboutit à la définition d'un modèle permettant la réalisation d'un système sur puce (SoC) pour la classification automatique des chromosomes.

3.3.1 Le diagramme en Y

Introduit par *Gajski* en 1983 [73], le diagramme en Y, illustré par la figure 3.3, est une représentation de la méthodologie de conception des circuits intégrés selon trois domaines de représentation permettant de modéliser les étapes de cette conception. Ces trois domaines sont partagés en cinq niveaux d'abstraction, plus ou moins élevés, à savoir le niveau système, le niveau algorithmique, le niveau transfert de registre, le niveau logique et le niveau circuit.

3.3.1.1 Le domaine comportemental

Ce domaine représente le fonctionnement du circuit ou du système, décrit sans introduire les ressources matérielles. Ce niveau d'abstraction engendre, principalement les spécifications d'un système, autrement dit le comportement de l'application qui est exprimé par des fonctions. Ces fonctions décrivent le calcul à réaliser et les relations entre entrées et sorties. La projection de la description du fonctionnement du circuit correspond à la transition du domaine comportemental au domaine structurel.

3.3.1.2 Le domaine structurel

Dans le domaine structurel le système est décomposé en sous-systèmes (objets architecturaux) interconnectés. Ce domaine fait le lien, via un processus automatisé de synthèse physique, entre le domaine comportemental et le domaine physique.

3.3.1.3 Le domaine physique

Ce domaine représente l'architecture physique du circuit ou le dessin de masque (*layout*). Il s'agit d'une représentation géométrique de la représentation structurelle; qui correspond à une synthèse physique.

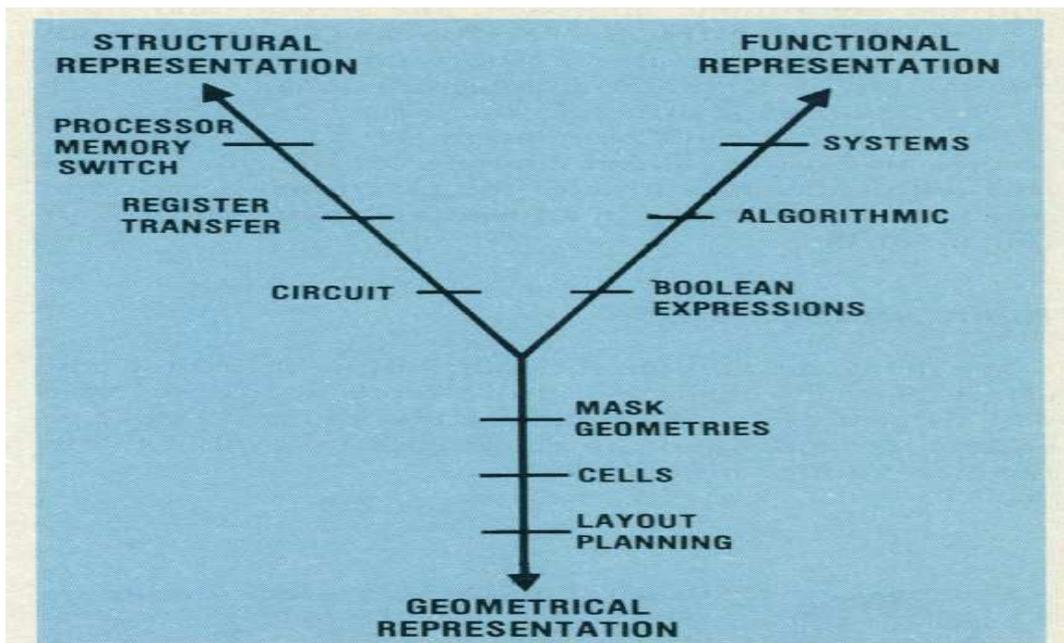


Figure 3.3 Le diagramme en Y [73]

3.4 Approche de conception du système de classification des chromosomes humains sur FPGA

L'approche adoptée est déduite du flot de conception d'un SoC; en passant des spécifications relatives à un AKS vers une implémentation hardware. Notre approche de conception est basée sur les considérations citées dans la section précédente et le flot de conception hardware ciblant principalement les circuits de type FPGA. Toutefois le modèle architectural proposé est indépendant de la technologie et peut être implémenté sur une autre technologie telle que la technologie ASIC (essentiellement celle basée sur les *standard cells*). Nous avons opté pour une architecture hardware du AKS en modélisant les différentes phases le constituant par des blocs architecturaux décrits en HDL (VHDL). Nous présentons dans ce qui suit, les phases de notre approche.

3.4.1 Choix du modèle ANN

Notre principal objectif est l'implémentation hardware de la partie classification qui représente la phase fondamentale dans un système de classification des chromosomes. Plusieurs travaux ont été effectués dans notre laboratoire [11-13] pour la classification des chromosomes avec des implémentations softwares. Ils ont traité, en particulier le modèle MLP, cependant nous visons une implémentation hardware qui doit répondre à certaines contraintes de surface utilisée, de consommation de puissance et de vitesse de calcul. C'est pourquoi nous avons choisi le modèle de Kohonen, qui nécessite moins de temps de calcul dans les phases d'apprentissage et de classification. En outre, le classificateur Kohonen donne des résultats comparables à ceux d'un classificateur MLP. Le choix du réseau de neurones de Kohonen est justifié, principalement, par:

1. Le coût en surface: La taille du réseau est l'une des considérations majeures dans une implémentation hardware d'un ANN. En effet, le nombre de neurones qui seront synthétisés sur le support hardware de type FPGA dépend de la taille du réseau, ce qui a une conséquence sur le coût en surface (utilisation des ressources). Dans la topologie MLP, il existe une couche intermédiaire supplémentaire par rapport au modèle de Kohonen, puisque le réseau MLP inclut des couches cachées (une ou

plusieurs couches avec un certain nombre de neurones additionnels) entre la couche d'entrée et la couche de sortie. La topologie du modèle de Kohonen n'a pas de couche cachée, donc le nombre de neurones dans l'architecture du réseau est réduit, ainsi la surface utilisée est réduite.

2. Le temps de traitement: L'apprentissage dans le modèle MLP est basé sur l'algorithme de rétro-propagation (*BP : Backpropagation*), qui est complexe comparé au modèle de Kohonen. Le problème avec le BP réside dans la complexité du processus d'apprentissage qui augmente le temps de traitement. Par conséquent, la convergence du réseau requière plus de temps. L'apprentissage dans l'algorithme de Kohonen est moins complexe et requière moins de temps de calcul; ce qui réduit le temps de convergence.

En conséquence le modèle de Kohonen est le plus approprié pour une implémentation hardware. La topologie choisie pour le classificateur est de type multi-réseau qui combine deux ANNs de type Kohonen. Le premier réseau classifie les chromosomes par leurs caractéristiques géométriques (L et IC) et le deuxième réseau les classifie en fonction de leurs densités de profil.

3.4.2 Choix du support hardware

Nous avons ciblé un support hardware de type FPGA pour notre implémentation, le choix d'une plateforme FPGA est justifié par le cycle de développement relativement court et la rapidité de traitement qu'assurent ces architectures reconfigurables. En plus les familles des FPGAs proposées sur le marché offrent une grande densité d'intégration, permettant d'implémenter tout un système sur une seule carte.

3.4.3 Choix du langage

Nous avons choisi le langage HDL puisqu'il s'agit dans notre approche d'un système n'incluant que des parties matérielles. En plus du parallélisme dans l'exécution, le HDL permet aussi la réutilisation des modules décrits dans d'autres blocs, cet aspect de réutilisation donne la possibilité d'optimiser le modèle déjà conçu. Une description HDL définit les fonctionnalités indépendamment de la technologie, l'architecture ainsi définie peut être implémentée sur FPGA ou ASIC ou autre support

hardware. Le langage HDL répond aux expressions du modèle en Y, par la définition du modèle comportemental ou fonctionnel en décrivant les fonctions du système à implémenter. Un modèle structurel est généré par l'interconnexion des modules élémentaires déjà décrits avec des descriptions comportementales. Une première validation du système conçu est envisageable avec des simulations fonctionnelles.

3.4.4 Définition du modèle architectural du SoC-AKS

L'étape suivante est la définition d'un modèle architectural, qui divise le système à implémenter en sous-systèmes. Cette phase est une phase essentielle dans notre approche, puisqu'elle permet de transformer la spécification initiale en une solution architecturale désirée. C'est à ce niveau que la décision concernant l'architecture du circuit permettant l'implémentation du SoC-AKS, principalement la partie classification basée sur le réseau de Kohonen, se fait. La solution architecturale proposée est une architecture modulaire constituée d'un ensemble de blocs fonctionnels interconnectés.

Le concept est de modéliser les phases d'un AKS en architectures hardwares permettant d'implémenter chaque étape comme sous-système du SoC-AKS. Les principales étapes permettant d'établir un modèle architectural pour les différents sous-systèmes sont résumées dans l'organigramme de la figure 3.4.

Le flot part d'une spécification système présentant les détails des opérations à effectuer; cette étape décrit ce que devra réaliser le système. Ces spécifications fonctionnelles seront ensuite transformées en un modèle architectural constitué d'un ensemble de blocs fonctionnels, afin de fixer la solution architecturale qui sera réalisée. L'étape suivante est la conception de cette solution; pour ce faire, chaque bloc fonctionnel est décrit en langage VHDL après avoir spécifié sa fonction dans l'ensemble (les spécifications de chaque bloc, ainsi que la fonction qu'il réalise seront détaillées dans le chapitre suivant).

Chaque bloc architectural modélisant un sous-système (une étape dans le AKS) est composé d'un nombre de modules interconnectés selon sa fonction. Ces sous-systèmes seront ensuite simulés et testés. Après avoir décrit, simulé et validé les différents blocs conçus de façon indépendante, on peut passer au modèle structurel. Cette étape consiste à interconnecter les blocs fonctionnels qui constituent notre SoC,

et donc à rassembler les sous-systèmes, afin de réaliser le système dans sa globalité, i.e. un AKS hardware.

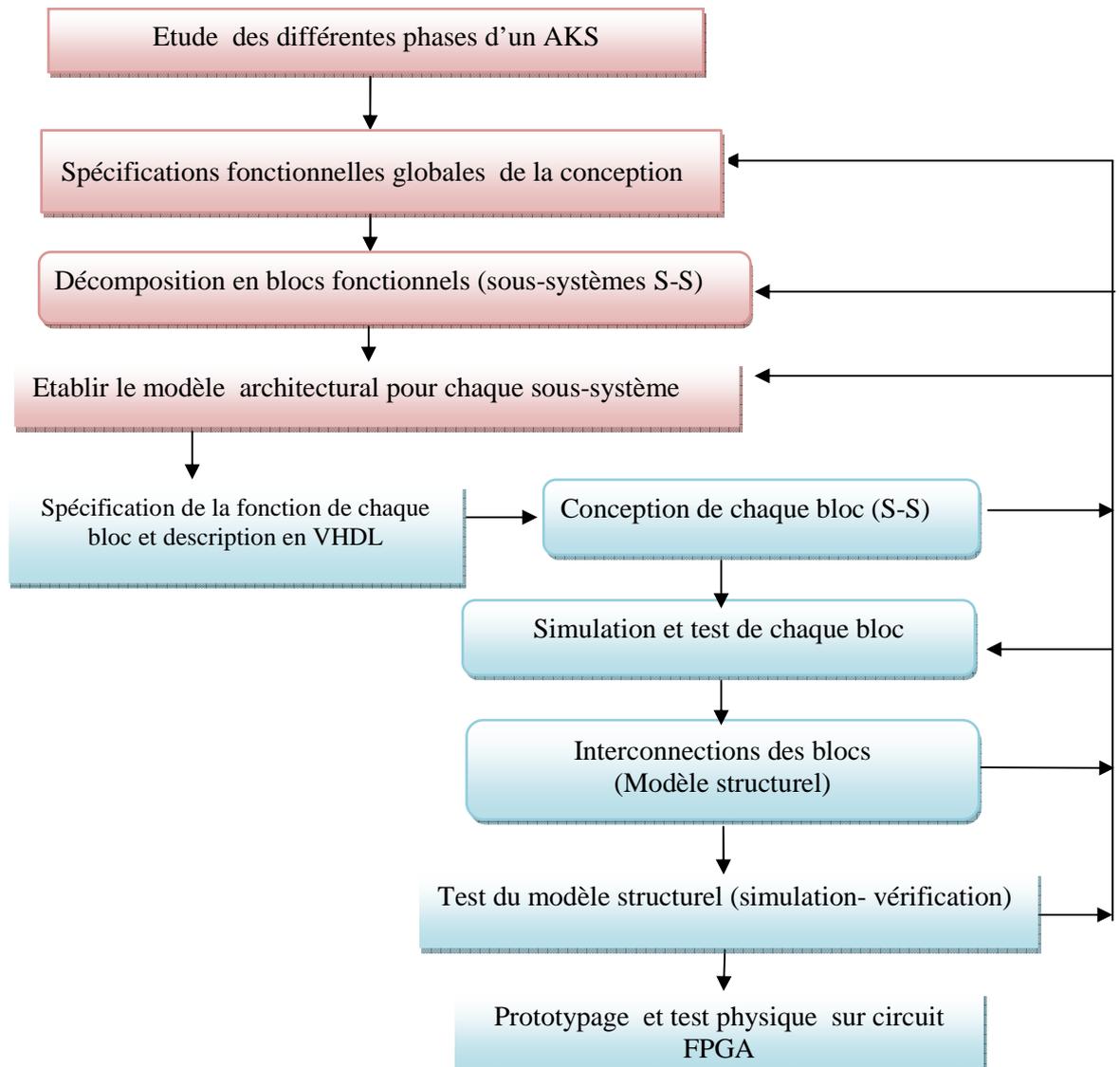


Figure 3.4 Flot de conception du SoC-AKS

L'avant dernière étape du flot de conception est la simulation du modèle proposé dans sa forme structurelle. Après validation et synthèse des modules conçus, l'étape d'implémentation est effectuée automatiquement en utilisant les outils d'implémentation qui permettent la génération de la *netlist* qui décrit les composants et leurs interconnexions sur le support hardware.

La dernière étape est la génération du fichier de configuration pour la programmation (configuration) de la carte FPGA, et le test du système sur le support physique

3.5 Définition des modèles architecturaux modélisant les différents sous-systèmes

3.5.1 Définition du modèle architectural modélisant la phase de traitement d'image

L'organigramme de la figure 3.5 résume les différentes étapes de la phase de traitement de l'image. Cette dernière inclut le prétraitement de l'image et la segmentation. L'étape prétraitement est relativement la plus simple comparée aux autres. Ce prétraitement est élaboré pour faciliter la segmentation et l'identification des chromosomes. Il consiste à améliorer la qualité de l'image métaphasique par élimination des bruits, en utilisant des filtres, par exemple le filtre médian.

La segmentation constitue la deuxième étape dans la phase de traitement de l'image, en effet la segmentation est hybride; où l'image est d'abord binarisée par l'intermédiaire d'un algorithme de seuillage comme par exemple, le *Otsu thresholding* qui détermine automatiquement le seuil. Après binarisation un autre algorithme est appliqué pour individualiser les chromosomes de l'image, par exemple la segmentation par croissance de région. La dernière étape consiste à concevoir les blocs architecturaux modélisant le sous-système traitement d'image. A l'issue de cette phase les images des chromosomes sont prêtes pour la squelettisation et l'extraction des paramètres.

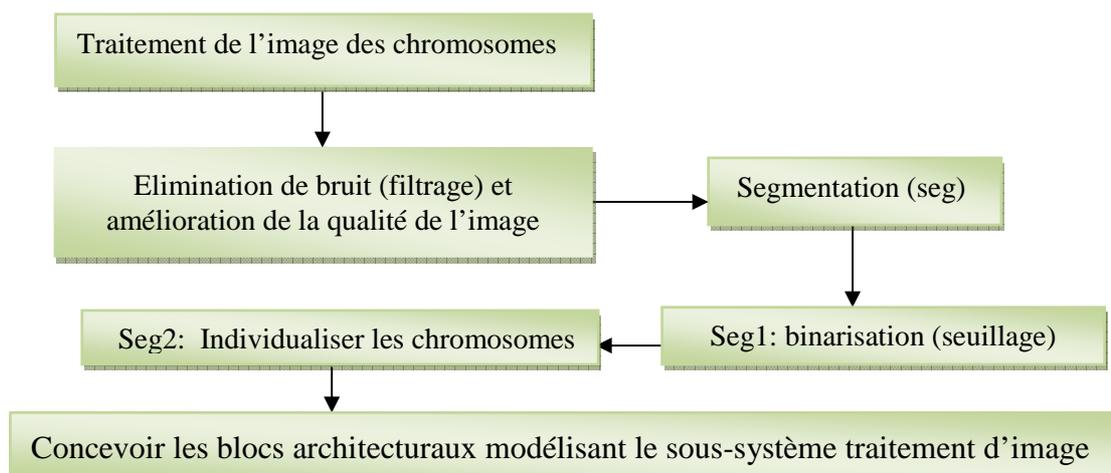


Figure 3.5 Flot de conception du sous-système traitement d'image

3.5.2 Définition du modèle architectural modélisant la phase extraction des caractéristiques

3.5.2.1 Méthodologie adoptée pour l'extraction des caractéristiques

Nous avons passé en revue, dans le chapitre 1, les différentes techniques d'extraction des caractéristiques, en particulier celles relatives à l'identification du centromère. L'estimation de l'axe médian avec la technique (MAT) [14, 20] est la première étape dans l'extraction des paramètres, il est utilisé pour le calcul de la longueur du chromosome (définie comme la somme des pixels constituant cet axe), et l'identification du centromère, ce dernier est considéré comme le plus petit segment parmi les segments passant par les pixels de l'axe médian du chromosome et perpendiculaire à ce dernier.

La deuxième technique utilisée pour la localisation du centromère est la projection vectorielle horizontale (PVH) [19, 21], qui est la somme des pixels de chaque ligne de l'image binarisée du chromosome. La PVH donne une meilleure exactitude comparée à la technique MAT pour les chromosomes courbés, selon l'étude rapportée dans [19].

Après étude des différentes techniques, nous avons opté pour une association de deux techniques, à savoir la technique MAT (en utilisant la squelettisation avec l'algorithme de Marthon) et la technique PVH. L'axe médian est utilisé pour le calcul des longueurs des chromosomes et les densités de profil. Pour l'identification du centromère et le calcul, par la suite, de l'indice centromérique nous avons choisi la technique PVH vu que cette dernière donne un meilleur résultat dans l'identification du centromère comparée à la MAT. Le centromère est considéré comme le minimum global dans le vecteur PVH. L'organigramme de la figure 3.6 résume les différentes étapes pour établir l'architecture hardware du sous-système extraction des caractéristiques.

Nous commençons par l'estimation de l'axe médian en utilisant la squelettisation avec l'algorithme de Marthon, ce dernier s'avère le plus approprié pour l'estimation de l'axe médian du chromosome, comme introduit précédemment. Les caractéristiques géométriques (L et IC) et les DPs sont calculées, par la suite. La dernière étape est la mémorisation des caractéristiques calculées, pour les utiliser dans la phase de classification.

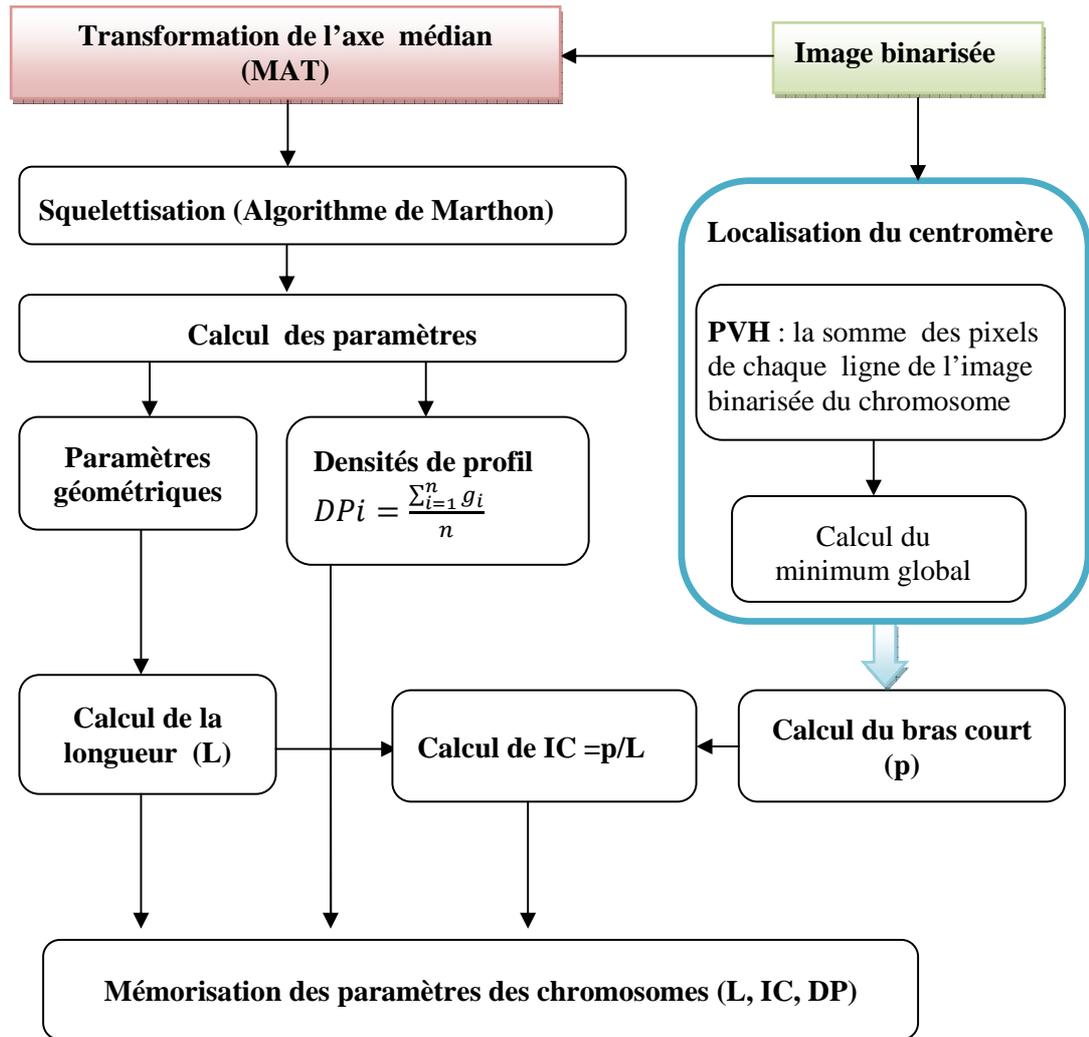


Figure 3.6 Les étapes de la méthodologie adoptée pour l'extraction des paramètres

3.5.2.2 Calcul des caractéristiques

A. Localisation du centromère et calcul de l'indice centromérique (IC)

Chaque point dans la PVH est obtenu par la somme des pixels de la ligne considérée de l'image binarisée du chromosome. Etant donné que l'image binarisée contient des pixels blancs (1) et des pixels noirs (0), la somme des pixels dans chaque ligne est la somme des pixels blancs. Le centromère est la région la plus étroite (localisée dans la région centrale) dans la structure du chromosome et donc la ligne avec le plus petit nombre de pixels blancs. En conséquence la somme de ces pixels correspond au minimum global (dans la région centrale) dans le vecteur PVH.

Cette étape est requise pour le calcul de l'IC qui est le rapport du bras court du chromosome (p) sur sa longueur totale (L). L'IC est calculé selon la formule (3.1).

$$IC = \frac{p \text{ (longueur du bras court du chromosome)}}{L \text{ (longueur totale du chromosome)}} \quad (3.1)$$

B. Calcul des densités de profil

Les densités de profil permettent de représenter les bandes chromosomiques qui se placent parallèlement le long de l'axe médian. Les vallées dans la courbe de profil indiquent les parties les plus claires de l'image du chromosome, tandis que les pics représentent des parties plus sombres.

Pour déterminer les valeurs DP_i , des segments de droites sont tracés le long de l'axe médian, passant par chaque pixel (p_i) de cet axe et perpendiculaire à celui-ci, comme illustré par la figure 3.7. Les segments de droite en question sont les points appartenant à la perpendiculaire à l'axe médian et au chromosome.

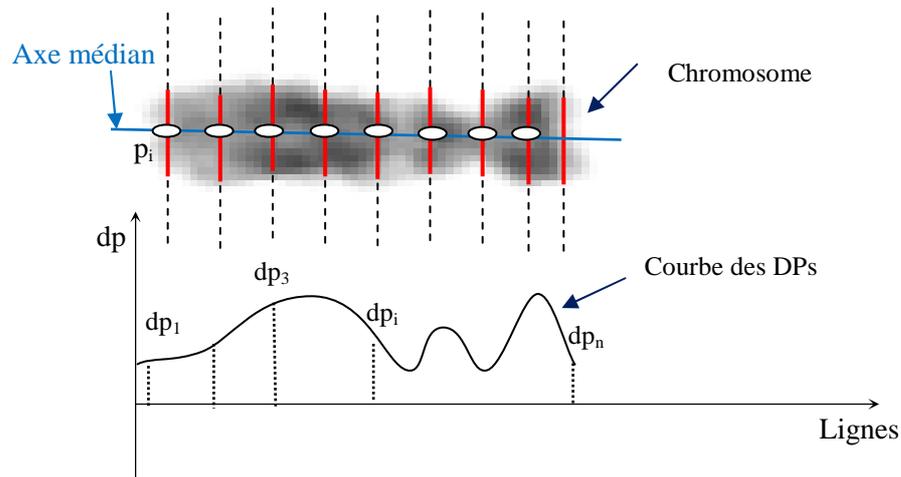


Figure 3.7 La courbe des densités de profil du chromosome

Chaque valeur de la densité de profil est la moyenne des intensités (les niveaux de gris) des pixels de la droite passant par le pixel (p_i) et perpendiculaire à l'axe du chromosome en ce point. DP_i est calculée selon la formule (3.2).

$$DPi = \frac{\sum_{i=1}^n g_i}{n} \quad (3.2)$$

Où n représente le nombre de pixels du segment de droite perpendiculaire à l'axe du chromosome et g_i , l'intensité du pixel (p_i).

Le nombre des densités de profil d'un chromosome est égal au nombre de pixels de son axe médian (son squelette), ce nombre est différent d'un chromosome à l'autre. Une discrétisation et normalisation [11] des courbes des densités de profil est effectuée pour trouver les vecteurs des densités de profil dont la taille est fixe pour tous les chromosomes.

3.5.3 Définition du modèle architectural modélisant la phase de classification

Le classificateur en question est basé sur le modèle de Kohonen. La modélisation du sous-système classification a été faite, principalement en deux étapes:

1. Etude de l'algorithme qui décrit le réseau de neurones de type Kohonen, qui se résume, principalement, en deux phases, à savoir :
 - La phase d'apprentissage qui se fait en deux étapes (la détermination du neurone gagnant et la mise à jour des poids synaptiques)
 - La phase de classification.
2. Translation des équations de Kohonen en blocs architecturaux.

3.5.3.1 Description de l'algorithme de Kohonen

L'algorithme de Kohonen, développé par T-Kohonen en 1984 [74] est un réseau à compétition dont la topologie est celle d'une carte auto-organisatrice ou SOM (*Self Organizing Map*). Les neurones sont organisés, généralement dans un réseau unidimensionnel ou bidimensionnel. La figure 3.8 illustre un réseau bidimensionnel dans une topologie carrée; cette dernière est la topologie communément utilisée. La couche d'entrée contient les vecteurs à classer et la couche de sortie contient des neurones qui rentrent en compétition dans la phase d'apprentissage (de type non supervisé). La compétition se fait dans un voisinage autour d'un neurone gagnant (également appelé *Best Matching Unit* (BMU)).

La notion de voisinage entre les neurones fait la particularité de ce réseau. Plus de détails sur l'algorithme de Kohonen sont donnés en annexe.

L'algorithme de Kohonen est relativement simple, il consiste en l'initialisation des vecteurs poids, la détermination du neurone gagnant, et la phase de mise à jour des poids synaptiques. Ces deux dernières phases sont régies par des équations que nous avons traduites en architecture hardware.

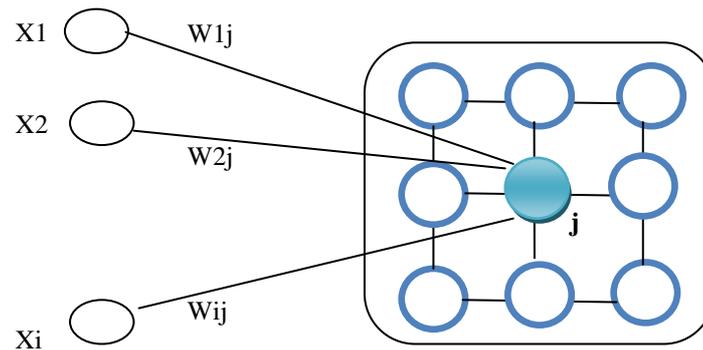


Figure 3.8 Carte de Kohonen

3.5.3.2 Flot de conception du sous-système classification

L'organigramme de la figure 3.9 résume les différentes étapes que nous avons suivies pour établir l'architecture hardware de ce classificateur. La phase d'apprentissage se fait en deux étapes: Le calcul de la distance et l'identification du neurone gagnant, et l'étape de mise à jour des poids synaptiques. La phase d'apprentissage commence par l'initialisation aléatoire des vecteurs de poids. La dimension du vecteur de poids de chaque neurone dans le réseau est égale à la dimension du vecteur d'entrée. L'initialisation appropriée consiste à choisir aléatoirement de petites valeurs différentes de W_j^0 , $j \in [1, J]$ où J est le nombre de neurones dans la carte.

Pour chaque vecteur d'entrée $X = (x_1, x_2, \dots, x_i)$, la distance à tous les vecteurs de poids de la carte est calculée; afin de déterminer le neurone gagnant qui est le neurone le plus proche, et donc celui avec la plus petite distance (son vecteur de poids est le plus similaire au vecteur d'entrée présenté). Communément la distance Euclidienne est utilisée pour le calcul de la distance (implémentation software), nous

avons considéré la distance de Manhattan dans notre architecture, vu que nous ciblons une implémentation hardware. Cette distance est calculée selon la formule (3.3).

$$d_j = \sum_{i=1}^n |x_i - w_{ij}| \quad (3.3)$$

Le neurone gagnant (j^*) est déterminé selon le critère de la distance minimale exprimé par (3.4).

$$j^* = \arg \min \|x_i - w_{ij}\| \quad (3.4)$$

Après détermination du neurone gagnant, son vecteur de poids et les vecteurs poids des neurones dans son voisinage sont ajustés au vecteur d'entrée pour être organisés en régions. Le processus de mise à jour est une variation de l'emplacement des neurones, c'est-à-dire que si le neurone est à l'intérieur du voisinage, ses poids seront ajustés, sinon sa position reste inchangée. La nouvelle valeur du poids (à l'itération "n + 1") est la fraction d'une différence entre le vecteur d'entrée et le vecteur de poids, qui est sommée au vecteur poids initial (à l'itération "n") du neurone considéré pour l'adaptation. La mise à jour des poids se fait selon la formule (3.5).

$$W_{ij}^{n+1} = W_{ij}^n + \eta^n h_n^{jj^*} (x_i^n - W_{ij}^n) \quad (3.5)$$

$$i \in [1, I], j \in [1, J]$$

Où η^n est le taux d'apprentissage et $h_n^{jj^*}$ est la fonction de voisinage, exprimée par (3.6). Elle définit l'influence du neurone gagnant (j^*) sur son neurone voisin (j). La fonction de voisinage diminue avec la distance entre la position du neurone gagnant (j^*) et la position du neurone voisin (j) dans la carte.

$$h_n^{jj^*} = \exp\left(-\frac{\|j - j^*\|^2}{2\sigma_n^2}\right) \quad (3.6)$$

σ : est le coefficient de voisinage; la décroissance de la taille du voisinage s'obtient par diminution de ce paramètre.

Etude de l'algorithme de Kohonen



Translation des équations de Kohonen (calcul de la distance et mise à jour des poids) en blocs architecturaux

Figure 3.9 Flot de conception du sous-système classification à base de l'algorithme de Kohonen

Dans la phase de classification, le neurone gagnant (le neurone le plus proche d'une entrée représentée par le vecteur X) est considéré comme l'indice correspondant au chromosome (sa classe) dont les caractéristiques sont représentées par le vecteur d'entrée X . Ce vecteur est assigné à la classe du neurone gagnant; notons que dans la phase de classification, les vecteurs poids sont ceux calculés dans la phase d'apprentissage (les poids mis à jours)

3.6 Conclusion

Nous avons décrit dans ce chapitre notre approche méthodologique pour l'implémentation hardware d'un système de classification des chromosomes humains. Notre objectif est d'élaborer une méthodologie permettant le passage des spécifications d'un AKS vers une implémentation hardware de type système sur puce ciblant un support hardware de type FPGA.

Notre approche de conception s'appuie essentiellement sur le flot de conception SoC et sur les considérations relatives aux spécifications d'un AKS. Nous avons exposé également, l'ensemble des étapes de notre démarche pour la solution hardware proposée; en justifiant les choix effectués à chaque étape.

Le principal choix effectué est relatif au modèle du réseau de neurones le plus approprié pour une implémentation hardware. Le choix est fait pour le modèle de Kohonen vu qu'il requière moins de temps de calcul, et donc moins de surface sur un support hardware.

Les architectures hardware modélisant le sous-système classification basé sur le modèle de Kohonen et le sous-système extraction des caractéristiques seront exposées dans le chapitre suivant.

4

Modélisation et implémentation hardware

4.1 Introduction

Plusieurs travaux ont traité la classification des chromosomes à base de réseau de neurones. La plupart des travaux effectués dans ce domaine consistent en des implémentations softwares. Pour exploiter le parallélisme des réseaux de neurones, une implémentation hardware et modulaire est proposée. Nous exposerons dans ce chapitre l'architecture du SoC proposé pour une implémentation sur une plateforme FPGA d'un AKS, en particulier le sous-système finalisé, à savoir le classificateur à base de réseau de Kohonen.

4.2 Architecture globale

4.2.1 Diagramme bloc du système proposé

La figure 4.1 illustre la structure globale du système proposé (une représentation simpliste de ses composants), ce dernier est composé des blocs suivants: Le bloc traitement d'image, qui inclut deux sous-blocs: le prétraitement et la segmentation de l'image. Le second bloc est le bloc extraction des caractéristiques, la squelettisation des images de chromosomes est effectuée en premier lieu pour

l'extraction de l'axe médian et le calcul des caractéristiques. Le vecteur des caractéristiques (L, IC, DP) est l'entrée du bloc classification. Ce dernier est basé sur une combinaison de deux réseaux de type Kohonen. Les détails des architectures des sous-systèmes, extraction des caractéristiques et classification seront exposés dans ce qui suit.

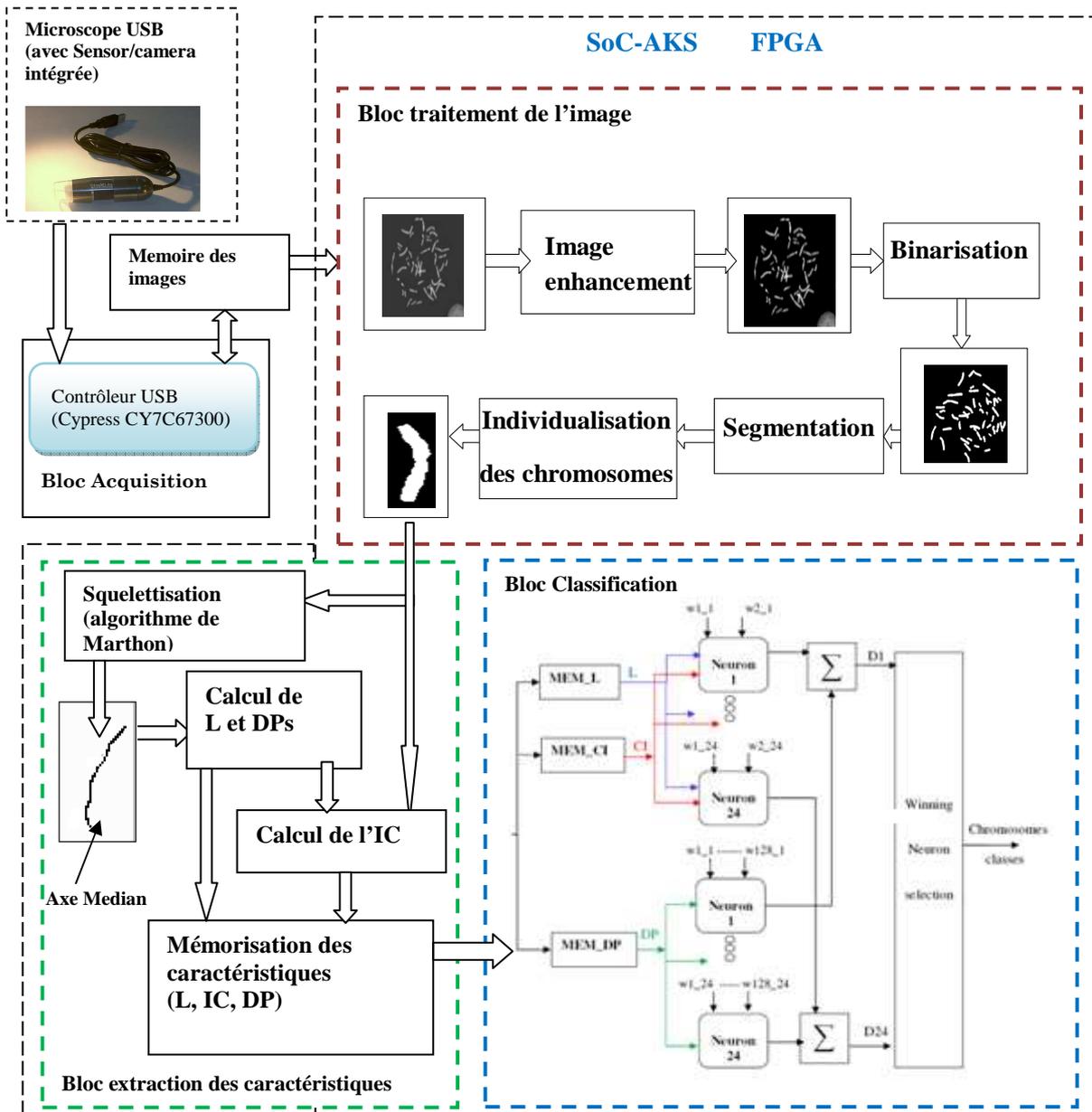


Figure 4.1 Schéma bloc du SoC-AKS proposé

4.2.2 Présentation de l'architecture hardware du sous-système classification

Le classificateur proposé comporte deux réseaux de type Kohonen, le premier réseau classe les chromosomes par leurs caractéristiques géométriques (L et IC) et le deuxième réseau les classe en fonction de leurs densités de profil. Pour les deux réseaux, la carte de Kohonen doit avoir une taille égale aux 24 classes possibles des chromosomes.

Le diagramme bloc de l'architecture proposée pour modéliser le sous-système classification est représenté sur la figure 4.2. L'architecture est composée d'unités (neurones) qui sont au nombre de 24 pour chacun des deux réseaux; ces unités correspondent aux 24 classes des chromosomes. Les composantes du vecteur de caractéristiques (L, IC, DP) sont stockées dans des mémoires MEM_L, MEM_IC et MEM_DP, respectivement. Ces caractéristiques sont les entrées de notre classificateur, et les indices qui correspondent aux paires de chromosomes représentent ses sorties. Les indices sont donc compris entre 1 et 24; chaque indice correspond à une classe, les chromosomes sexuels X et Y sont indiqués par les indices 23 et 24, respectivement.

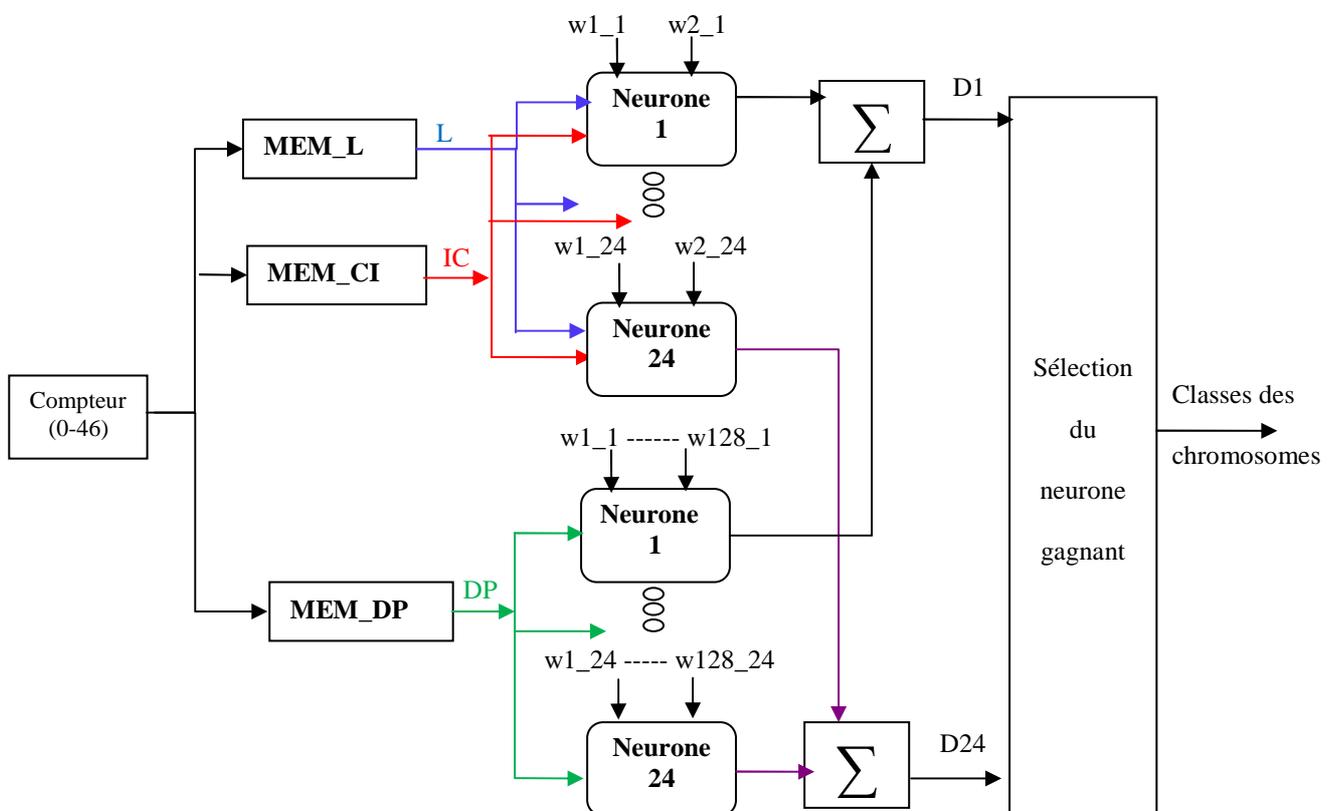


Figure 4.2 Diagramme bloc de l'architecture du sous-système classification

4.3 Implémentation des deux phases de l'algorithme de Kohonen

Afin de pouvoir implémenter l'algorithme de Kohonen en hardware, nous avons procédé à l'élaboration des blocs architecturaux modélisant les deux phases de cet algorithme, à savoir la phase d'apprentissage et la phase de classification. Les architectures de ces deux blocs sont exposées dans ce qui suit. Dans cette section, nous exposons l'architecture du classificateur selon les caractéristiques géométriques (L et IC), le deuxième réseau est conçu de la même manière.

4.3.1 Implémentation de la phase d'apprentissage

Dans la première étape de la phase d'apprentissage, la distance de chaque vecteur d'entrée à tous les vecteurs poids est calculée afin de déterminer le neurone gagnant (BMU). Communément, la distance Euclidienne est utilisée pour le calcul de la distance et l'identification du neurone gagnant. Cependant l'implémentation de la distance Euclidienne en hardware consomme plus de ressources et de puissance, en raison de la racine carrée et des multiplications requises pour implémenter la puissance 2. Ainsi d'autres distances sont considérées pour l'implémentation hardware de l'étape de calcul de la distance, afin de réduire le temps de calcul et la taille du circuit. Notamment la distance de Manhattan qui est utilisée dans plusieurs implémentations hardware du *SOM* [75,76].

Nous avons opté pour la distance de Manhattan, pour l'implémentation de la première étape de l'algorithme de Kohonen, afin de simplifier le calcul et réduire la surface utilisée. La figure 4.3 représente le schéma bloc de l'architecture qui modélise la phase d'apprentissage; elle est constituée de trois modules principaux:

Le bloc mémorisation des paramètres, le bloc détermination de BMU (l'unité calcul de la distance et l'unité sélection du neurone gagnant) et le bloc mise à jour des poids synaptiques.

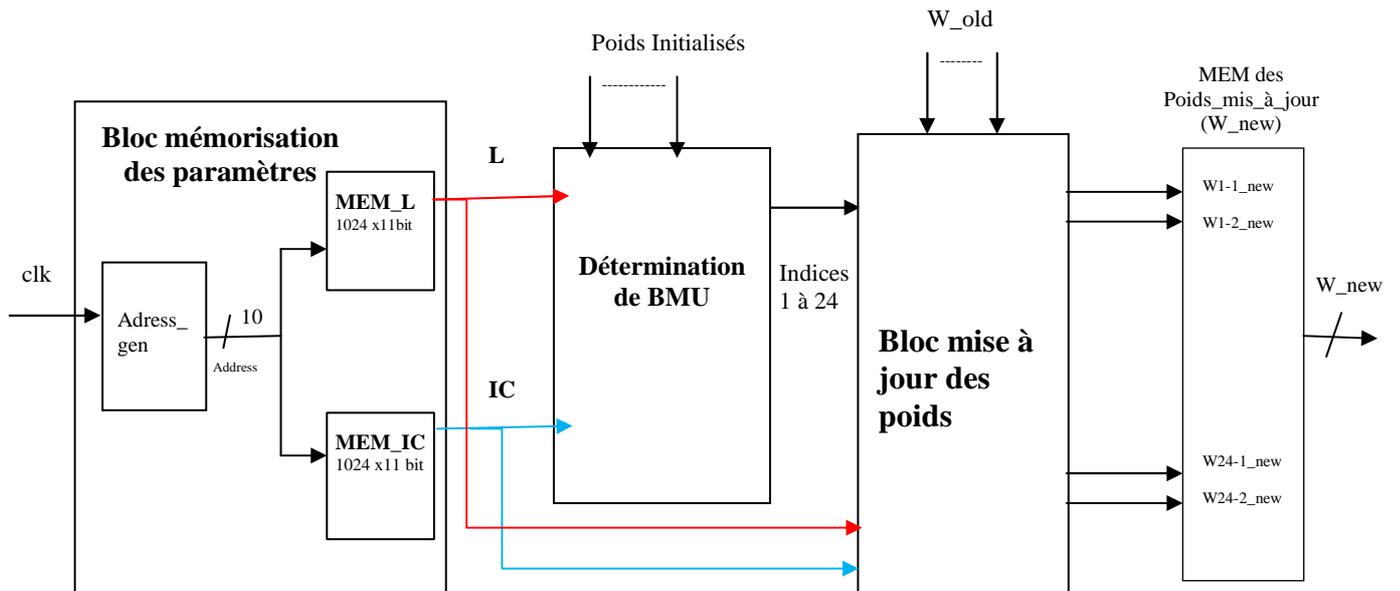


Figure 4.3 Schéma bloc de l'architecture de la phase d'apprentissage

4.3.1.1 Bloc mémorisation des paramètres

Ce module est composé de mémoires et un générateur d'adresses. Les entrées sont les caractéristiques géométriques L et IC (les densités de profil pour le deuxième réseau) qui sont stockées dans des mémoires (MEM_L, MEM_IC) de taille (1024x11bit). Les caractéristiques et les poids synaptiques (W) sont codés sur 11 bits.

4.3.1.2 Bloc détermination de BMU

La figure 4.4 illustre l'architecture du bloc détermination de BMU qui est composé de deux unités: l'unité calcul de la distance de Manhattan (*Manhattan_dist*, une unité pour chaque neurone) et l'unité sélection du neurone gagnant.

a. Calcul de la distance

Les modules *Manhattan_dist* modélisent l'équation (4.1) de calcul de la distance de Manhattan entre l'entrée X présentée au réseau et les vecteurs de poids des neurones de la carte. Chaque neurone dans le réseau a un vecteur de poids $W_j = (W_{1j}, W_{2j}, \dots, W_{ij})$, ces vecteurs poids sont initialisés aléatoirement.

$$d_j = \sum_{i=1}^n |x_i - w_{ij}| \tag{4.1}$$

Le vecteur d'entrée du classificateur représente les caractéristiques géométriques ou les densités de profil. Le vecteur d'entrée pour les caractéristiques géométriques est $X = (L, IC)$ tandis que le vecteur d'entrée pour les densités de profil est $X = (DP_1, DP_2, \dots, DP_{128})$.

L'architecture de cette unité est relativement simple; elle est composée de soustracteurs et d'un accumulateur; cette unité est utilisée pour chaque neurone et sa sortie (distance) est comparée aux autres distances dans l'unité sélection du neurone gagnant (qui compare les 24 distances). La figure 4.5 illustre l'architecture interne (schématique) de l'unité calcul de la distance du 1^{er} neurone.

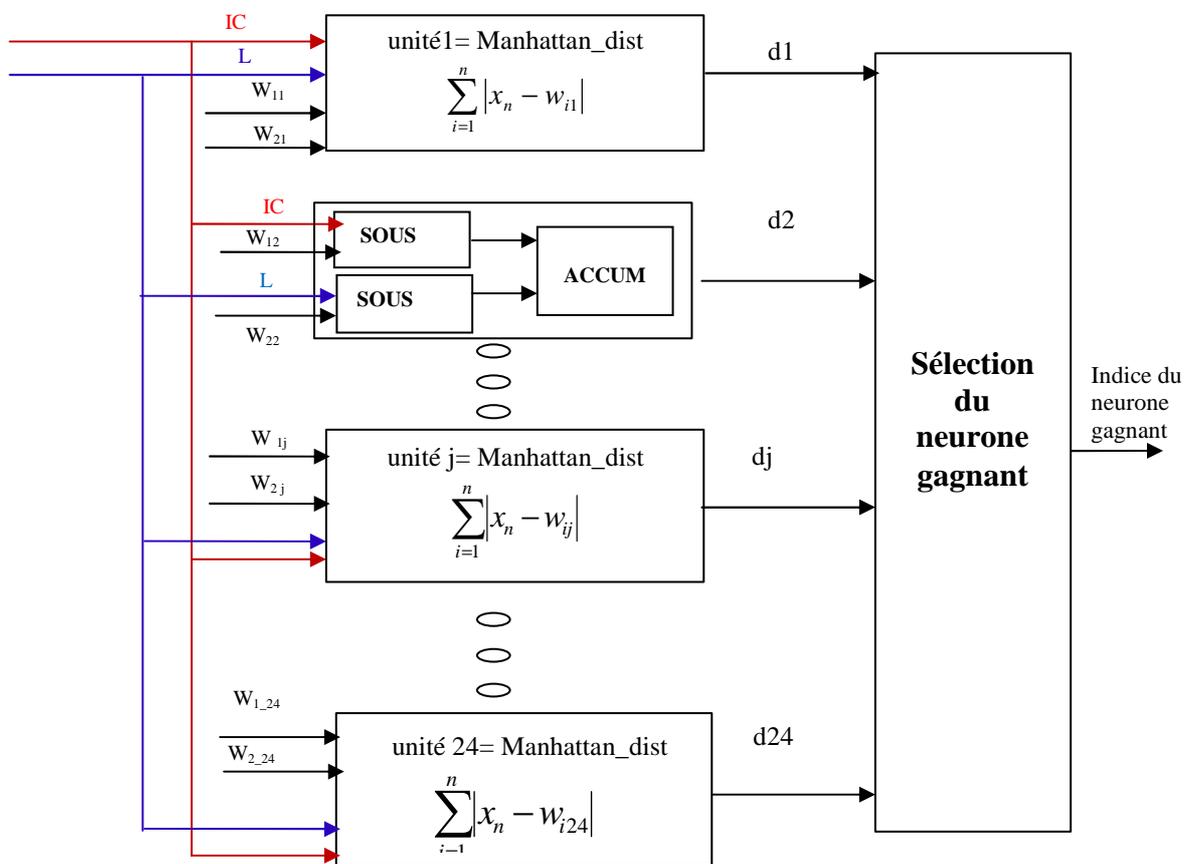


Figure 4.4 Architecture du bloc détermination de BMU

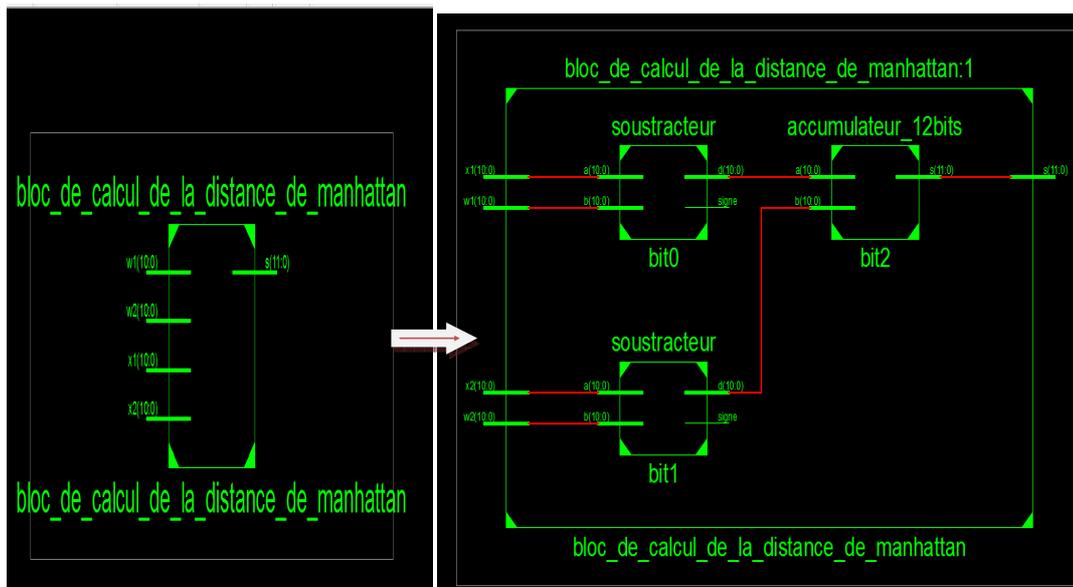


Figure 4.5 Architecture interne (schématique) du module calcul de la distance du 1^{er} neurone

b. Sélection du neurone gagnant

L'unité de sélection du neurone gagnant compare les distances, puis sélectionne la plus petite d'entre elles. Le neurone dont la distance (entre son vecteur poids W et l'entrée X) est la plus petite est sélectionné comme neurone gagnant. Cette unité est composée d'une succession de comparateurs arrangés dans une configuration en cascade. Le premier étage comprend trois modules de comparaison (4 comparateurs dans chaque module). Chaque module sélectionne la distance minimale entre 8 distances. La sortie de chaque comparateur (la distance minimale) alimente le comparateur adjacent pour être à nouveau comparée à une autre distance jusqu'à ce que la plus petite distance entre les 24 neurones soit trouvée.

Le deuxième étage comprend six comparateurs pour sélectionner le minimum entre les sorties des comparateurs du premier étage. Le troisième étage comprend 3 comparateurs et le dernier comprend un comparateur qui compare les trois distances. L'indice du neurone dont la distance est minimale (le neurone gagnant) est donné en sortie de cette unité. La figure 4.6 montre un exemple de comparaison de distances pour 8 neurones. Ce module est utilisé trois fois puisqu'on compare 24 distances (24 neurones en compétition). La figure 4.7 illustre l'architecture interne (schématique) de l'unité sélection du neurone gagnant et la figure 4.8 représente l'architecture interne du module BMU.

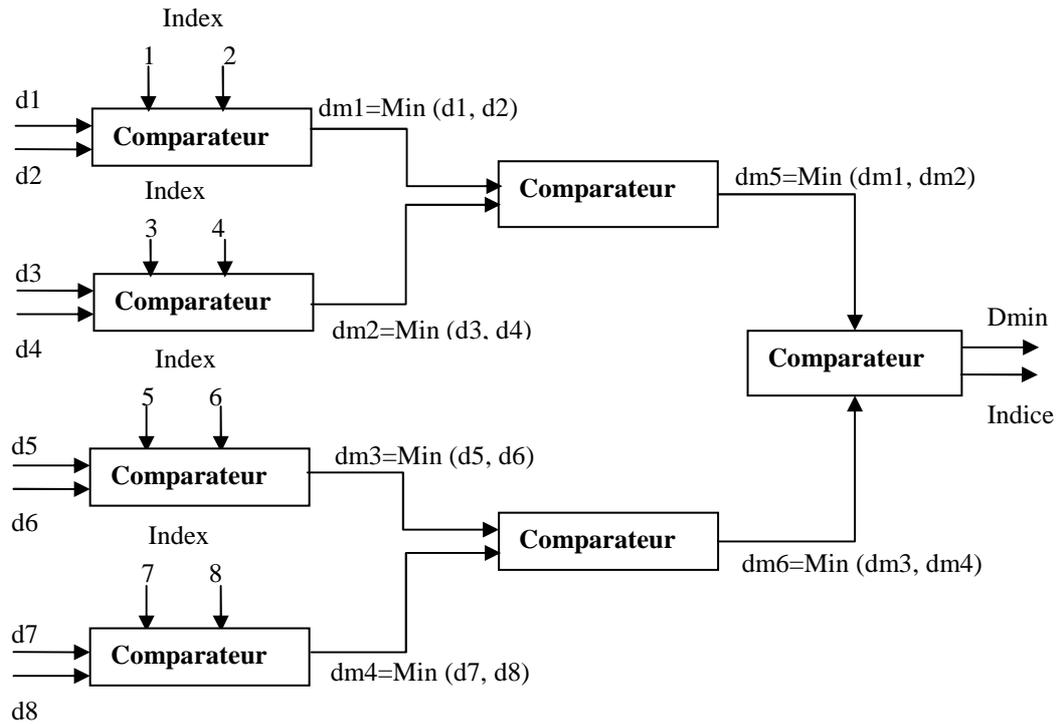


Figure 4.6 Comparaison de distance pour 8 neurones

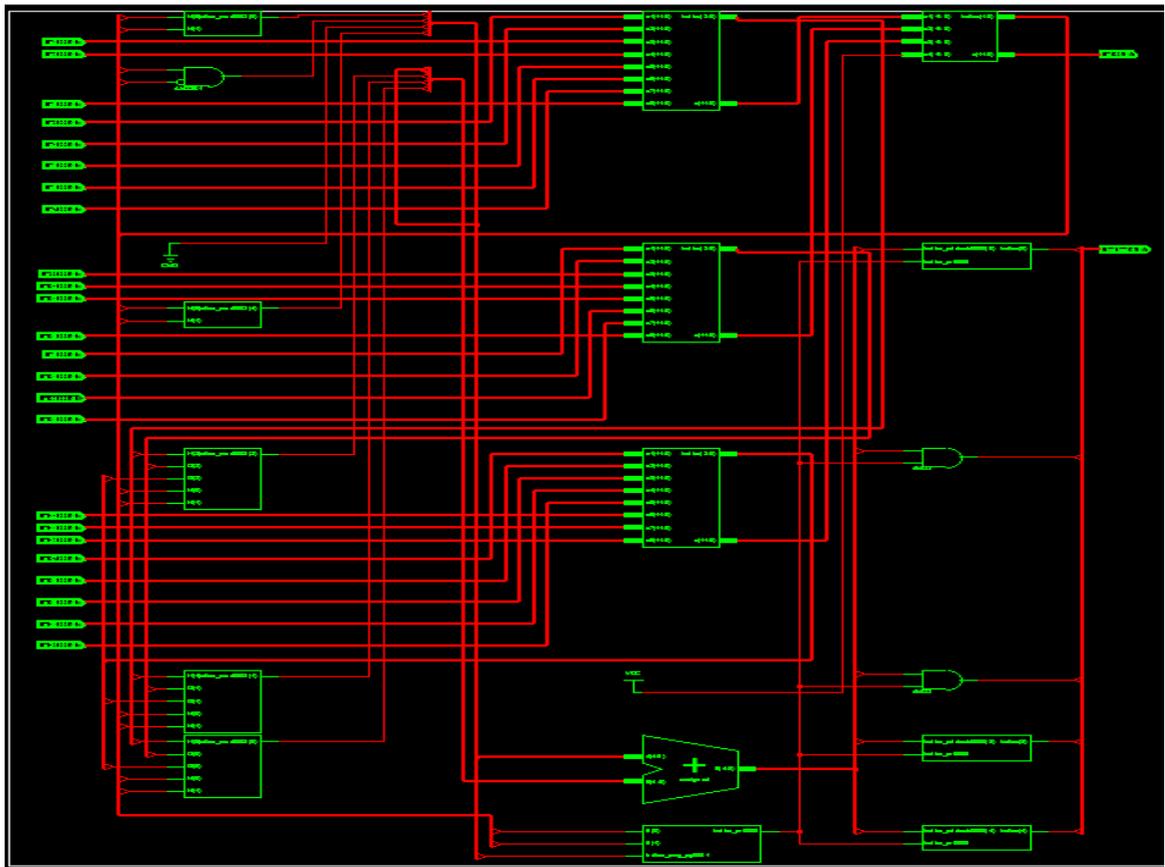


Figure 4.7 Architecture interne (schématique) du module sélection du neurone gagnant

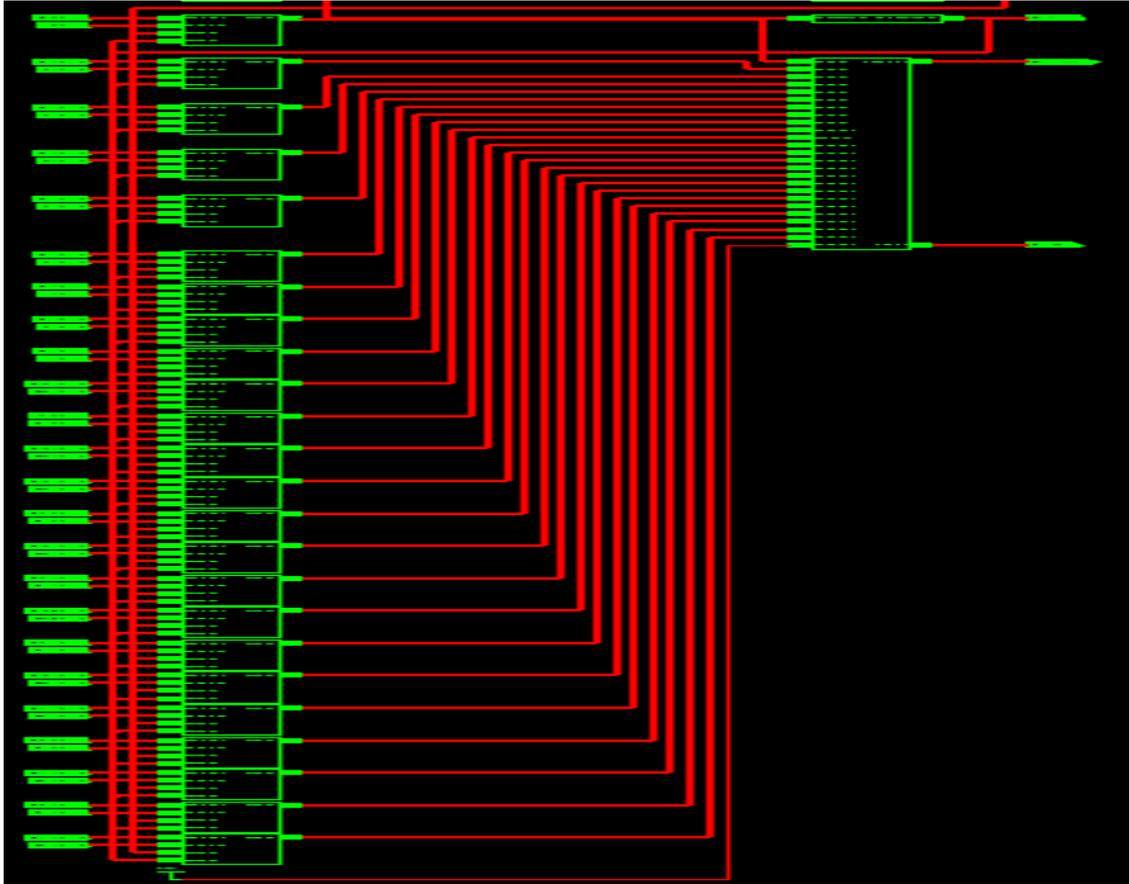


Figure 4.8 Architecture interne du bloc BMU

4.3.1.3 Bloc mise à jour des poids synaptiques

Ce module modélise l'équation de mise à jour des poids, le vecteur des poids du neurone gagnant (déterminé précédemment par le module sélection du neurone gagnant) et les vecteurs poids des neurones dans son voisinage sont mis à jour selon la formule (4.2).

$$w_{ij}(n+1) = w_{ij}(n) + \varphi(n) * (x_i - w_{ij}(n)) \quad (4.2)$$

$$\varphi(n) = \begin{cases} \alpha(n) & \text{Si } j = j^* \\ \alpha(n)H(n) & \text{Si } j \in N(j^*, r(n)) \text{ voisinage autour du neurone gagnant } (j^*) \\ 0 & \text{Autrement} \end{cases}$$

Le terme $\alpha(n)$, désigne le coefficient d'apprentissage; c'est une fonction décroissante. Ce terme est conçu avec un décompteur dans l'implémentation proposée. La valeur de $\alpha(n)$, est décrétementée à chaque itération; lorsque cette valeur est égale à '0', ce module renvoie un signal d'arrêt (arrêt d'apprentissage).

$H(n)$ est une fonction décroissante avec la distance $d(j^*, j)$ entre la position du neurone gagnant (j^*) et la position du neurone voisin (j). La valeur de $\varphi(n)$, est déterminée en fonction de l'appartenance (ou pas) du neurone excité au voisinage du neurone gagnant.

1. Si le neurone considéré est le neurone gagnant, c'est-à-dire, $j^* - j = 0$ alors

$$\varphi(n) = \alpha(n)$$

2. Si le neurone considéré est à l'intérieur du voisinage du neurone gagnant alors

$$\varphi(n) = \alpha(n)H(n)$$

3. Si le neurone considéré n'est pas dans le voisinage du neurone gagnant alors

$$\varphi(n) = 0$$

Le bloc de mise à jour des poids de chaque neurone est composé de 2 unités de calcul (UC) et du module sélection des nouvelles valeurs des poids. La figure 4.9 représente l'architecture interne (schématique) du module de mise à jour des poids du $j^{\text{ème}}$ neurone. Le module de sélection fournit les nouvelles valeurs des poids en fonction du critère de sélection suivant:

1. Si le neurone n'est pas dans le voisinage du neurone gagnant, ses poids resteront inchangés.

2. Si l'indice d'entrée est égal à l'indice du neurone gagnant, la nouvelle valeur du poids est : $w_{ij}(n+1) = w_{ij}(n) + \alpha(n) * (x_i - w_{ij}(n))$

3. Si l'indice est égal à l'indice du neurone voisin, la nouvelle valeur du poids est: $w_{ij}(n+1) = w_{ij}(n) + \alpha(n) * H(n) * (x_i - w_{ij}(n))$

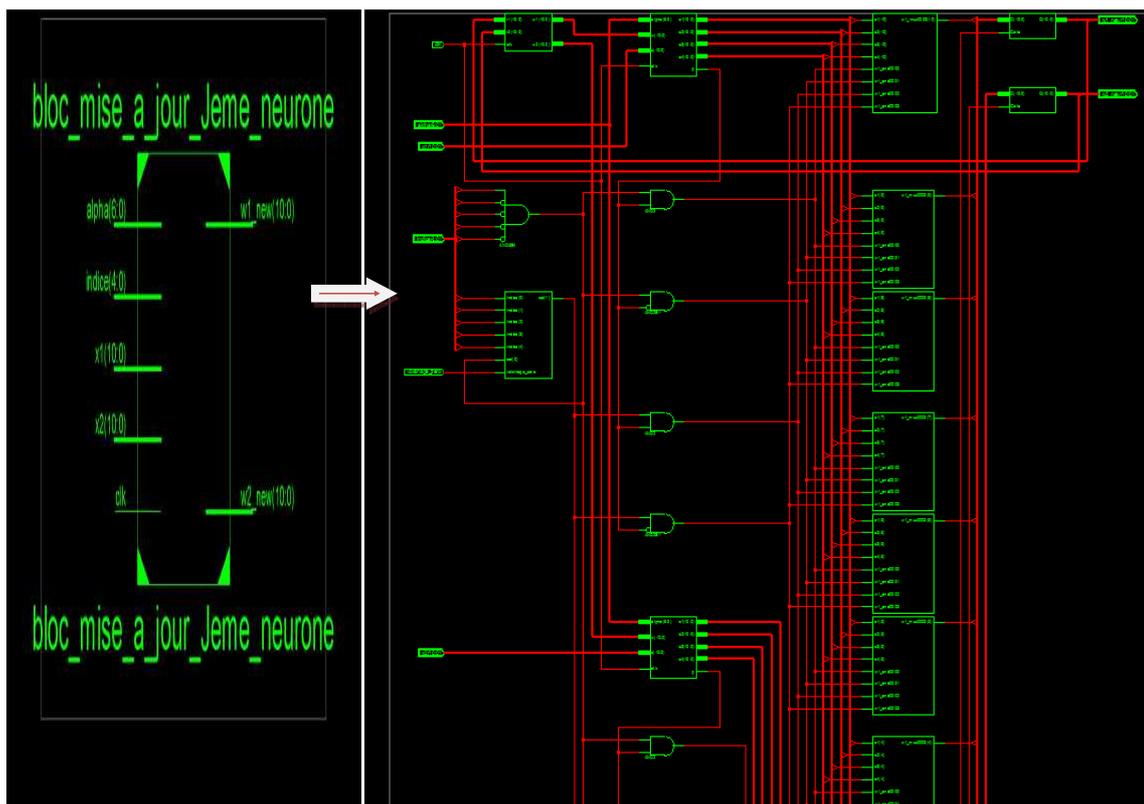


Figure 4.9 Architecture interne (schématique) du module de mise à jour des poids du $j^{\text{ème}}$ neurone

4.3.2 Implémentation de la phase de classification

Dans la phase de classification, l'architecture du bloc calcul de la distance et détermination du neurone gagnant est la même que celle de la phase d'apprentissage. La différence est dans la taille des mémoires. Ces dernières sont de tailles (46x11bits) comme illustré sur la figure 4.10 représentant le schéma bloc de l'architecture de la phase de classification. Les poids utilisés dans cette phase sont déterminés par le bloc de mise à jour des poids dans la phase d'apprentissage.

Les indices qui correspondent aux paires de chromosomes sont les résultats donnés en sorties. Dans cette phase, l'unité de sélection du neurone gagnant délivre l'indice du neurone avec la plus petite distance. Ce dernier spécifie la classe correcte du chromosome. Les résultats de la classification des chromosomes sont affichés sur l'écran LCD de la plateforme FPGA après implémentation physique.

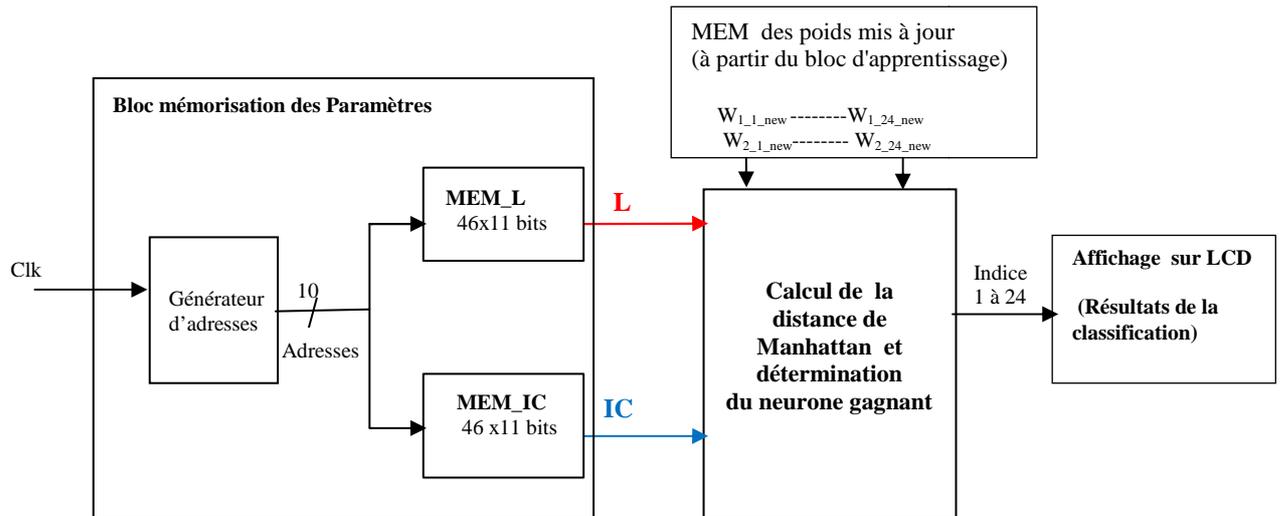


Figure 4.10 Schéma bloc de l'architecture de la phase de classification

4.4 Présentation de l'architecture hardware du sous-système extraction des caractéristiques

La figure 4.11 représente le schéma bloc de l'architecture hardware proposée pour le sous-système extraction des caractéristiques.

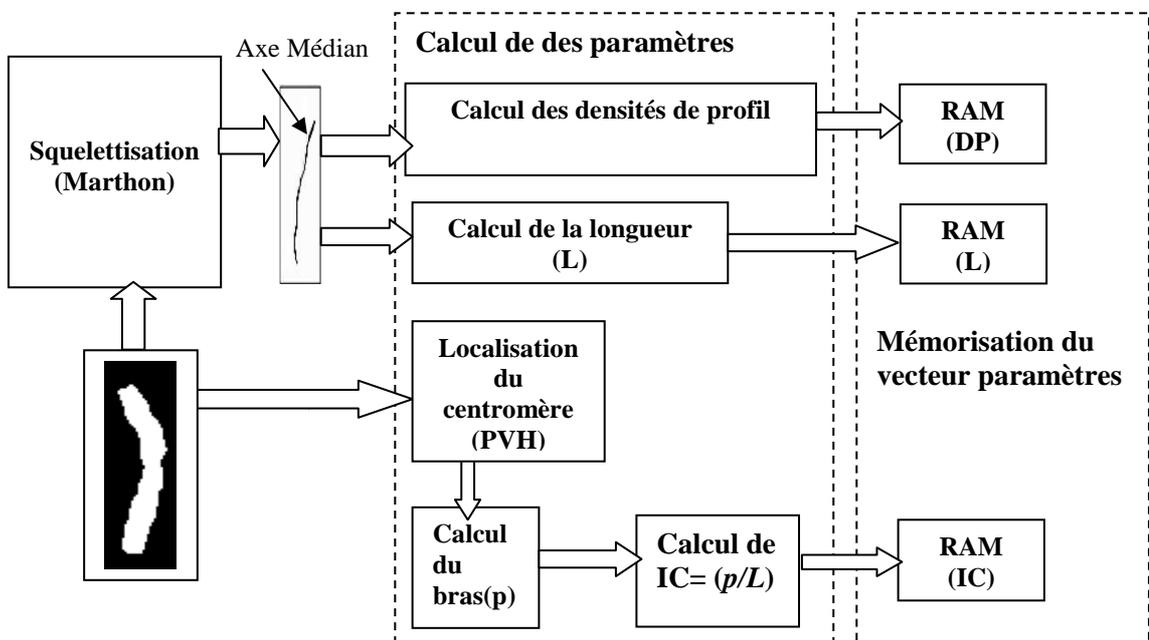


Figure 4.11 Schéma bloc de l'architecture hardware du sous-système extraction des caractéristiques

Comme expliqué, précédemment dans le chapitre 3, l'extraction des caractéristiques est basée sur une combinaison des techniques MAT et PV. Plus précisément la PVH qui sera employée pour la localisation du centromère et le calcul, par la suite, de l'indice centromérique. Le centromère est le minimum global dans la PVH. L'estimation de l'axe médian sera faite par la squelettisation avec l'algorithme de Marthon; les longueurs des chromosomes et les DPs sont par la suite, calculées dans les sous-blocs calcul de la longueur du chromosome et calcul de DP, respectivement.

4.4.1 Calcul des caractéristiques

Nous présenterons dans cette section la proposition des architectures pour les différents modules de calcul des caractéristiques.

4.4.1.1 Architecture du module PVH et localisation du centromère

Ce module calcule les valeurs du vecteur PVH et détermine la position du centromère (minimum global). Chaque point dans le vecteur PVH est obtenu par la somme des pixels de la ligne considérée de l'image binaire du chromosome (qui contient des pixels blancs (1) et des pixels noirs (0)). Le centromère correspond à la ligne avec le plus petit nombre de pixels blancs; la somme de ces pixels correspond au minimum global (dans la région centrale) du vecteur PVH. La figure 4.12 représente l'architecture de ce module, dont l'entrée est l'image binarisée du chromosome, cette dernière est stockée dans une RAM de taille (N (nombre de lignes de l'image) * M (nombre de colonnes de l'image)).

Les lignes de l'image binarisée sont mises dans un tableau de taille (M) valeurs des pixels, chaque ligne contient M pixels codés sur 8 bits. Les pixels de chaque ligne seront sommés dans un accumulateur qui calcule la somme $SOM(L_j) = \sum_{i=1}^M p_i$. Chaque $SOM(L_j)$ représente une valeur du vecteur PVH. Les sommes ($SOM(L_j)$, $j = [1, N]$) calculées seront par la suite enregistrées dans le tableau vecteur PVH et comparées pour déterminer le minimum global (dans la région centrale) du vecteur PVH, ainsi constitué.

La plus petite valeur comprise entre les valeurs $SOM(L_{N/4})$ et $SOM(L_{N/2+N/4})$ est sélectionnée comme le minimum global du vecteur PVH (Min_PVH), cette

dernière est la sortie du comparateur et correspond à la position du centromère. En effet nous comparons les sommes comprises entre $SOM(L_{N/4})$ et $SOM(L_{N/2+N/4})$, vu que le centromère est considéré comme le minimum global dans la région centrale du vecteur PVH. Par exemple pour une image dont le nombre de lignes est $N=224$, la région centrale sera entre L_{56} et L_{168} .

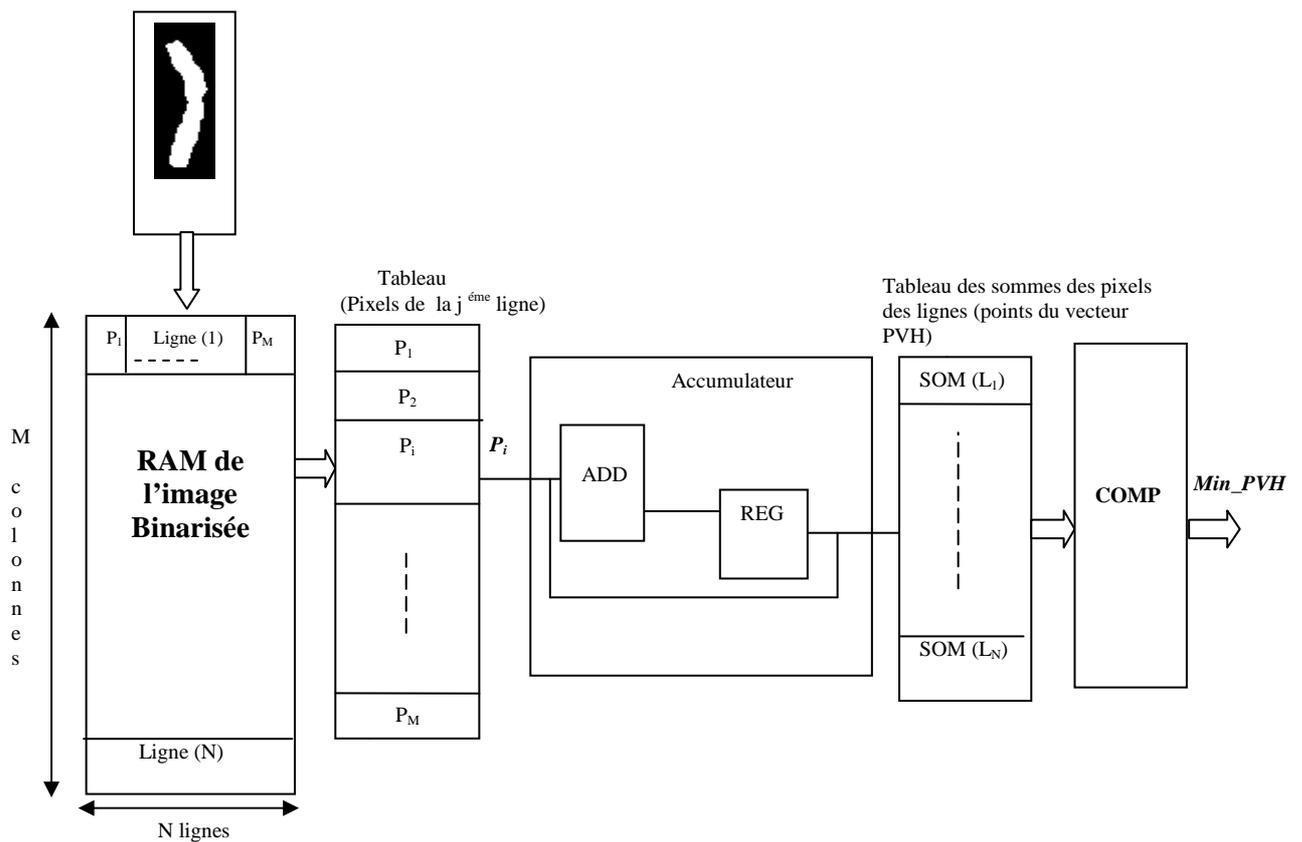


Figure 4.12 Architecture du module PVH et localisation du centromère

Le bras court du chromosome est la partie supérieure de celui-ci, comme illustré sur la figure 4.13. La longueur du bras court (p) correspond au segment limité par la ligne (L_1) et la ligne représentant le centromère c.-à-d. la ligne dont la somme des pixels correspond au Min_PVH .

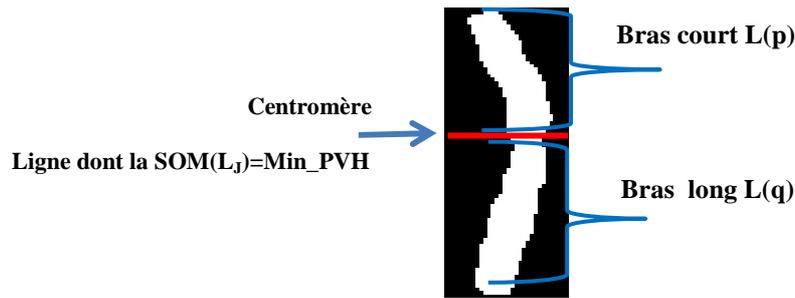


Figure 4.13 Centromère qui sépare le chromosome en deux bras

La figure 4.14 représente l'architecture interne (schématique) du module calcul des points du vecteur PVH.

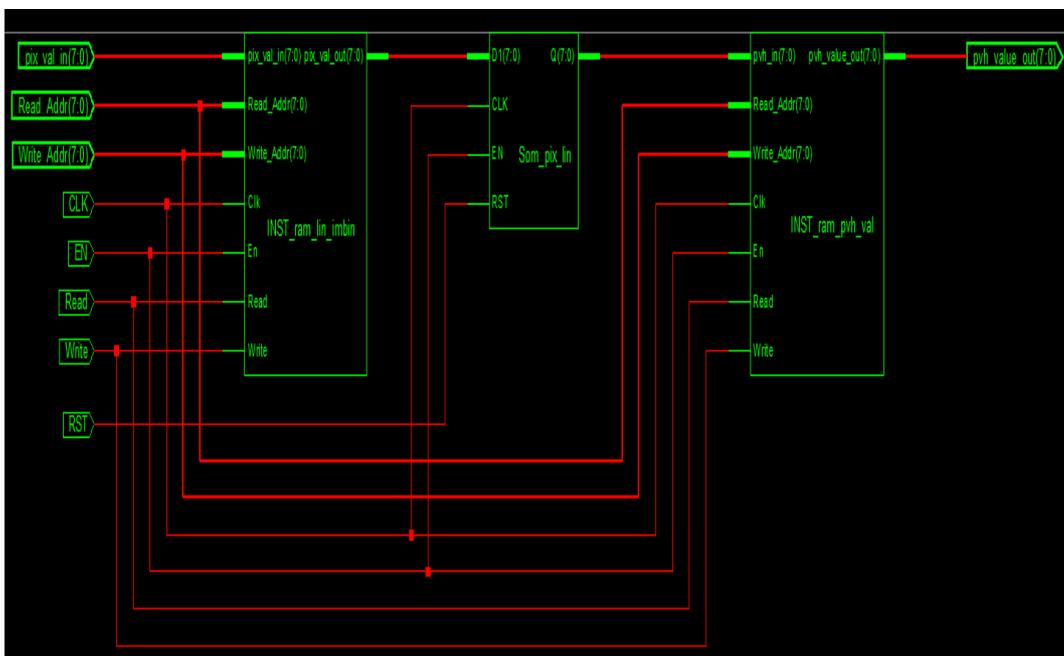


Figure 4.14 Architecture interne (schématique) du module calcul des points du vecteur PVH

4.4.1.2 Calcul de la longueur du chromosome

La figure 4.15 représente l'architecture du module calcul de la longueur du chromosome et la figure 4.16 représente son architecture interne (schématique). La longueur du chromosome est calculée en sommant les pixels de son axe médian, donc de son squelette. Pour avoir le nombre des pixels noirs et donc la longueur du chromosome, nous avons utilisé un compteur qui s'incrémente à chaque fois qu'un

pixel noir est trouvé. Ce compteur est contrôlé par un *enable*, qui est la sortie d'un comparateur comme illustré sur la figure 4.15.

L'image du squelette du chromosome est stockée dans une RAM, chaque pixel (codé sur 8bits) de l'image squelette est comparé à la valeur 0 (noir). La comparaison est effectuée dans le module comparateur dont la sortie servira de bit de sélection (*enable*) pour le compteur (Cnt+), si le pixel est noir le compteur est incrémenté si non la valeur reste inchangée. La sortie du compteur *Count_out (L)* représente le nombre de pixels noirs dans le squelette du chromosome et donc sa longueur, qui sera enregistrée dans la mémoire des longueurs RAM_Ls.

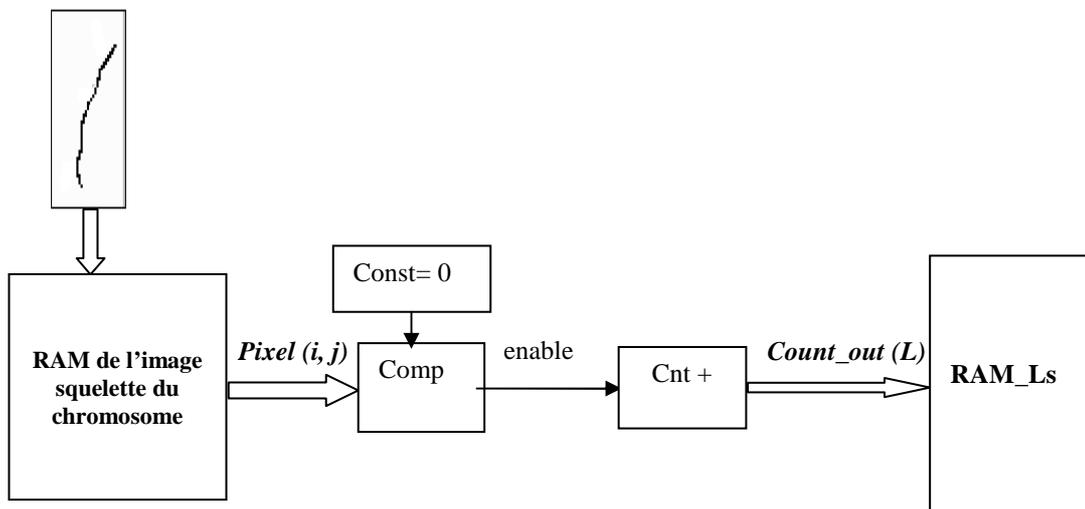


Figure 4.15 Architecture du module calcul de la longueur

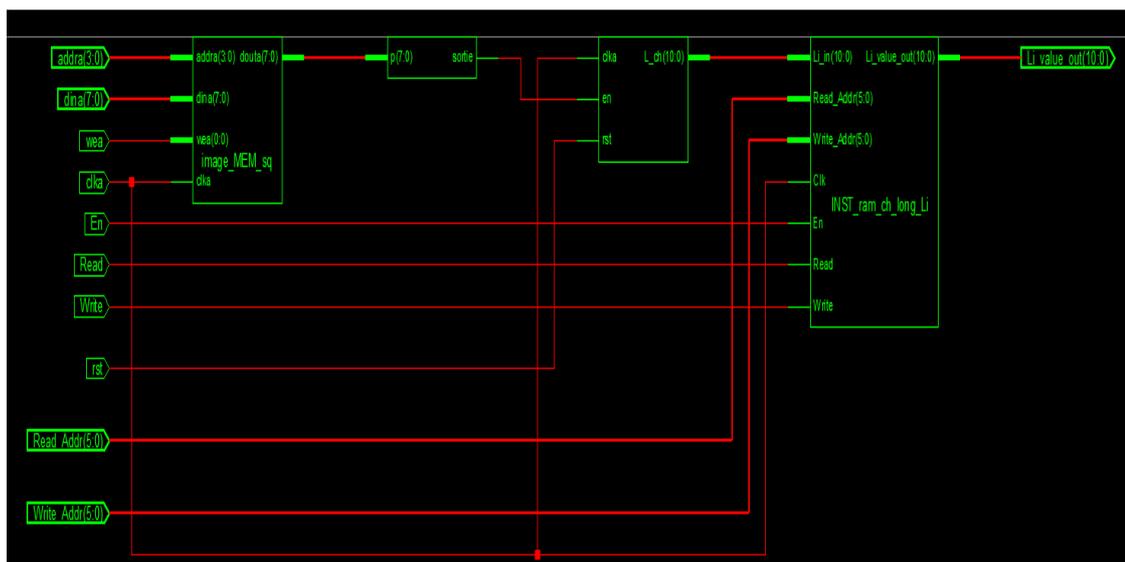


Figure 4.16 Architecture interne du module calcul de la longueur du chromosome

4.4.1.3 Calcul de l'indice centromérique

L'IC est calculé selon la formule (4.3)

$$IC = \frac{p \text{ (longueur du bras court du chromosome)}}{L \text{ (longueur totale du chromosome)}} \quad (4.3)$$

Ce module est implémenté avec un simple diviseur dont les entrées sont la longueur totale du chromosome (L) et la longueur du bras court (p), déterminée précédemment après localisation du centromère (dans le module PVH).

4.4.1.4 Calcul des densités de profil

Chaque valeur de DP est calculée selon la formule (4.4), DP_i est la moyenne des intensités des pixels de la droite passant par le pixel (p_i) et perpendiculaire à l'axe médian en ce point.

$$DP_i = \frac{\sum_{i=1}^n g_i}{n} \quad (4.4)$$

n : est le nombre des pixels du segment de droite perpendiculaire à l'axe du chromosome.

g_i : est l'intensité du pixel (p_i).

La figure 4.17 illustre l'architecture du module calcul de DP et la figure 4.18 représente son architecture interne (schématique). En effet il y a une similarité dans la conception de ce module et celui de la PVH. Dans les deux cas l'image du chromosome est traitée ligne par ligne, et un calcul des sommes est effectué en utilisant un accumulateur. La différence réside dans les valeurs à sommer, qui sont les intensités des pixels de chaque ligne dans le cas du module calcul de DP_i (les pixels dans le module PVH), et aussi dans le type d'image à traiter qui est l'image du squelette du chromosome (image binarisée du chromosome dans le module PVH).

Les intensités des pixels sont stockées dans une RAM (chaque ligne représente les intensités des pixels du segment de droite). Les intensités des pixels de chaque segment (chaque ligne (j) $j=[1, J]$) sont mises dans un tableau de 'n' valeurs (n

intensités), et sommées dans un accumulateur. Pour calculer la moyenne des intensités, les sommes ainsi calculées seront divisées par le nombre de pixels de la ligne considérée (n pixels du segment de droite). Un IP diviseur *IPcorgen* est utilisé pour effectuer cette division, chaque moyenne représente une densité de profil (DP_i).

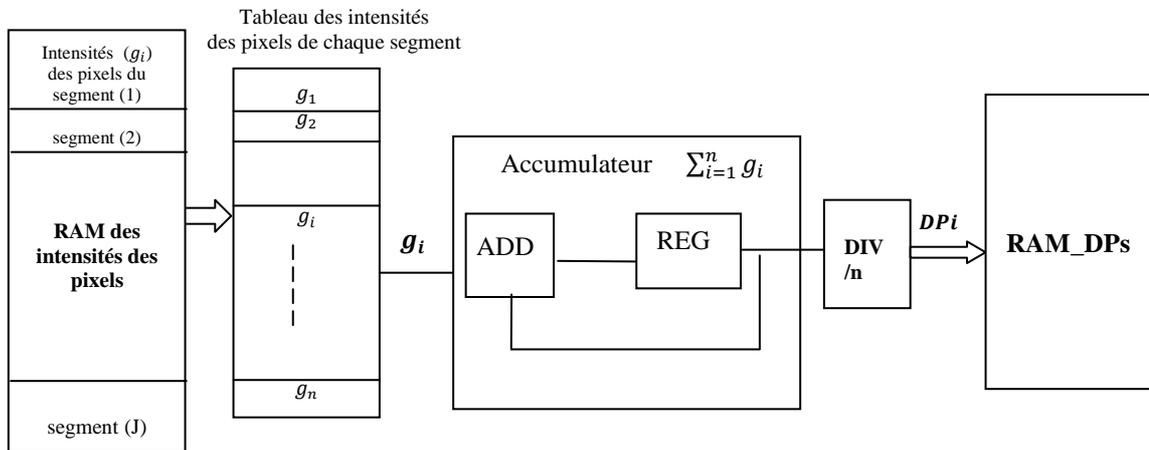


Figure 4.17 Architecture du module calcul de la DP

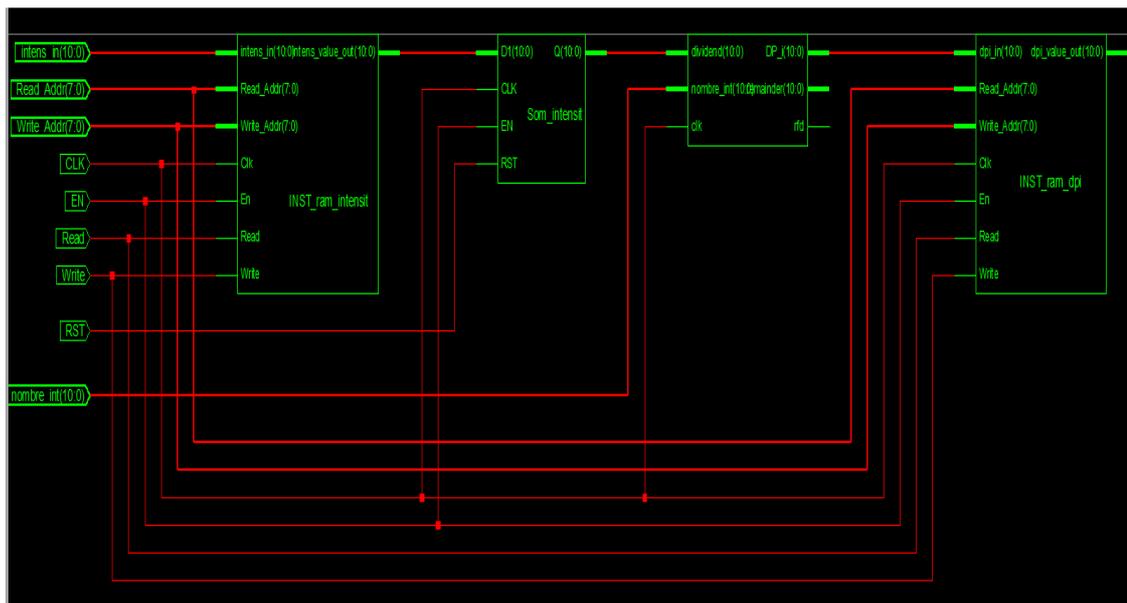


Figure 4.18 Architecture interne (schématique) du module calcul de la DP

Le dernier sous-bloc est constitué de mémoires, comme illustré sur la figure 4.11, pour stocker les caractéristiques (L, IC, DP) calculées.

4.5 Conclusion

Nous avons présenté dans ce chapitre le modèle architectural proposé pour l'implémentation hardware d'un AKS. Une architecture modulaire constituée de sous-systèmes représentant les étapes d'un AKS. Chaque sous système étant composé d'un ensemble d'IPs réutilisables.

Vu la complexité du SoC-AKS nous avons procédé à la conception et l'implémentation des sous-systèmes séparément, et étant donné que le sous-système classification est la partie prépondérante du SoC-AKS, nous avons commencé par la conception et l'implémentation de ce dernier. Les résultats de synthèse et implémentation de l'architecture du classificateur exposé dans ce chapitre ainsi que les discussions inhérentes feront l'objet du chapitre suivant.

5

Prototypage et résultats

5.1 Introduction

Nous présenterons dans ce chapitre les résultats de prototypage du sous-système finalisé, à savoir le sous-système classification. Nous exposerons, également des résultats de synthèses sur des plateformes plus récentes telles que la Virtex7 et Kintex7. Les résultats d'une évaluation des performances du classificateur proposé, en termes de taux de classification, de *Million-Connection-per-second* (MCPS) et de *Million-Connection-Update-per-second* (MCUPS), seront présentés par la suite.

Nous aborderons également dans ce chapitre quelques travaux qui ont traité des implémentations hardware du ANN de type Kohonen. Ce qui permettra de comparer les résultats obtenus pour le classificateur proposé avec ceux rapportés dans ces travaux en termes de MCPS et MCUPS.

5.2 Prototypage

5.2.1 Résultats de synthèse et implémentation

Le bloc classification a été prototypé, en premier lieu, en utilisant la plateforme de développement Spartan3E de Xilinx, pour une validation fonctionnelle de la conception. Les résultats de l'implémentation ciblant la Spartan3E sont rapportés dans nos travaux antérieurs [77-80]. La figure 5.1 illustre les résultats de la classification des chromosomes. Les indices (classes) qui correspondent aux paires de chromosomes sont affichés sur l'écran LCD de la plateforme FPGA après implémentation physique.

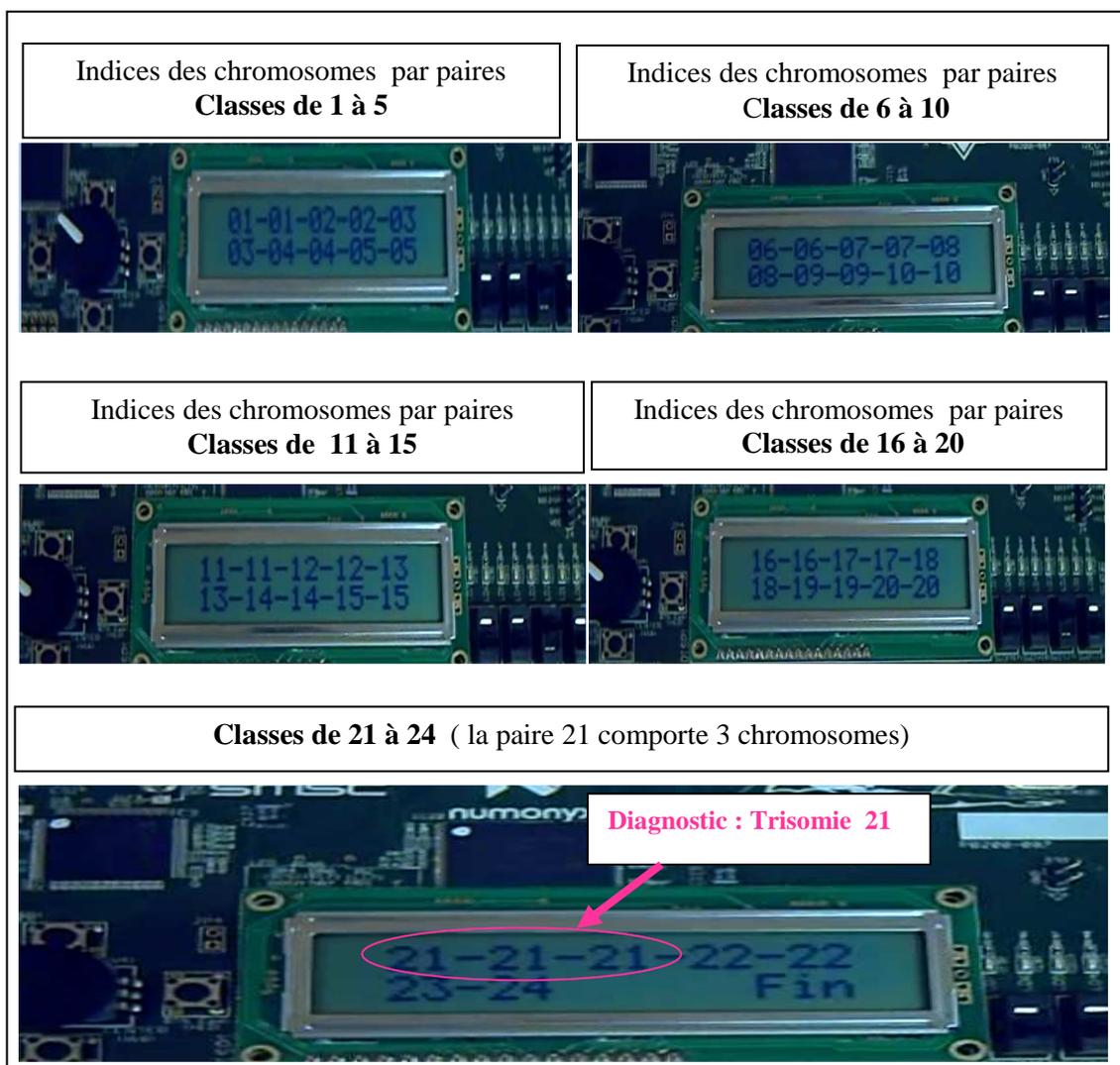


Figure 5.1 Résultats de la classification des chromosomes affichés sur l'écran LCD de l'FPGA

Comme le SoC-AKS inclut d'autres sous-systèmes et que la Spartan3E est limitée en ressources; une exploration de cartes cibles a été faite dans le but de choisir la famille du FPGA (avec plus de ressources) qui peut contenir le SoC-AKS dans sa globalité. Nous présentons dans les tableaux 5.1 à 5.5, les résultats de synthèse détaillés pour différentes famille de FPGAs. Le tableau 5.6 résume les résultats de l'implémentation pour ces FPGAs, en termes de ressources utilisées et de fréquence maximale.

Tableau 5.1 Résultats de synthèse Virtex 6

Device Utilization Summary (estimated values)				
Virtex6	Logic Utilization	Used	Available	Utilization
	Number of Slice Registers	1440	93120	1%
	Number of Slice LUTs	3326	46560	7%
	Number of fully used LUT-FF pairs	1351	3415	39%
	Maximum Frequency	89.38 MHz		

Tableau 5.2 Résultats de synthèse Virtex 7

Device Utilization Summary (estimated values)				
Virtex7	Logic Utilization	Used	Available	Utilization
	Number of Slice Registers	1441	357600	1%
	Number of Slice LUTs	3320	178800	1%
	Number of fully used LUT-FF pairs	342	3419	13%
	Maximum Frequency	104.99 Mhz		

Tableau 5.3 Résultats de synthèse Kintex 7

Device Utilization Summary (estimated values)				
Kintex7	Logic Utilization	Used	Available	Utilization
	Number of Slice Registers	1440	38000	3%
	Number of Slice LUTs	3317	19000	17%
	Number of fully used LUT-FF pairs	1342	3415	39%
	Maximum Frequency	87.59MHz		

Tableau 5.4 Résultats de synthèse Spartan 3E

Device Utilization Summary (estimated values) Spartan 3E			
Logic Utilization	Used	Available	Utilization
Number of Slices	2508	4656	53%
Number of Slice Flip Flops	1444	9312	15%
Number of 4 input LUTs	4876	9312	52%
Maximum Frequency	48.59MHz		

Tableau 5.5 Résultats de synthèse pour une Virtex5 (XC5VLX50-1FF676)

Ressources (Slices)	Ressources utilisées	Ressources disponibles	Utilisation %
# Slice Registers	1428	28800	4%
# Slice LUTs	3451	28800	11%
# LUT-FF pairs	1399	3480	40%
# bonded IOBs	9	440	2%
Fréquence Maximale	76.77 MHz		

Tableau 5.6 Résultats de synthèse pour différents circuits FPGAs (résumé)

FPGAs	% Utilisation de Slice Registers	% Utilisation de Slice LUTs	% Utilisation de LUT-FF pairs	Fréquence Max (Mhz)
Virtex 5	4%	11%	40%	76.77
Virtex 6	1%	7%	39%	89.38
Spartan3E	53%	52%	15%	48.59
Virtex 7	1%	1%	43%	104.99
Kintex 7	3%	17%	39%	87.59

L'outil de développement utilisé pour la conception et l'implémentation est l'outil ISE 13.1 de Xilinx, les résultats de synthèse et implémentation de la partie classification du SoC, ciblant une plateforme de type ML501-Virtex5, dont nous disposons au niveau de notre laboratoire, sont présentés dans le tableau 5.5, la fréquence de fonctionnement est de 76 MHz. Le sous-système implémentant la partie classification utilise 4% de *Slices Registres* et 11% de *LUTs*. Il est à noter que les *LUT* sont les ressources les plus utilisées, l'utilisation des autres ressources reste faible.

La figure 5.2 obtenue avec l'outil graphique *FPGA Editor* représente le *layout* de l'architecture du sous-système réalisé, à savoir le bloc classification implémenté sur une Virtex-5.

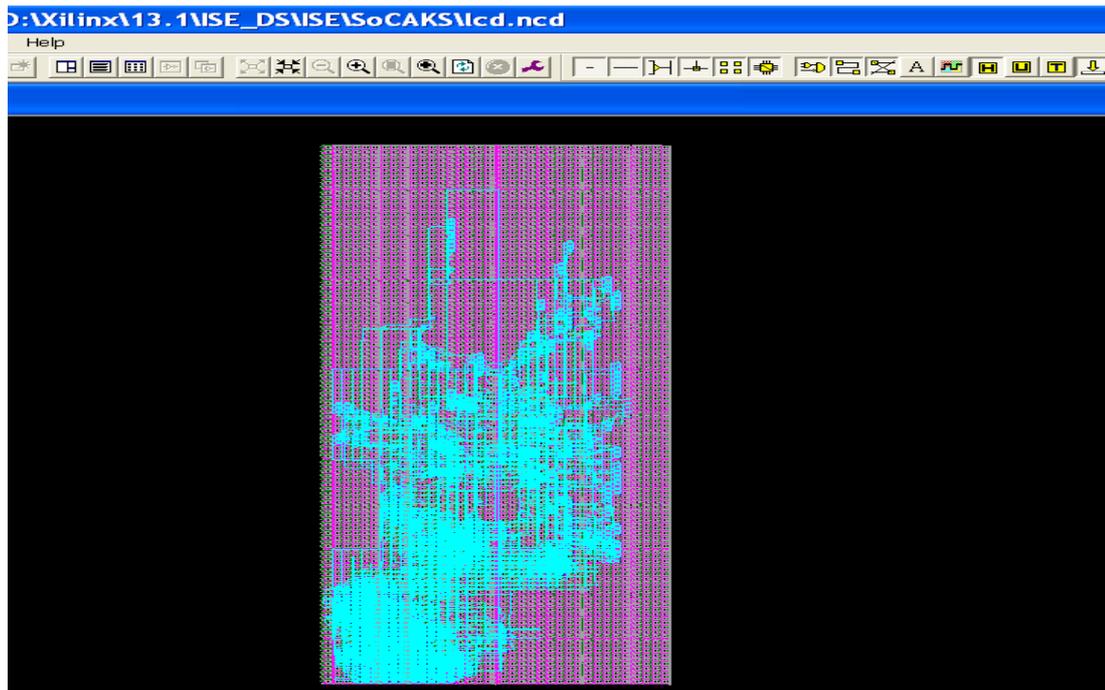


Figure 5.2 Layout de l'architecture du sous-système classification

5.2.2 Résultats en termes de puissance consommée

Les résultats en termes de puissance consommée sont présentés dans le tableau 5.7. L'estimation de la consommation de puissance a été faite en utilisant l'outil *XPower Analyzer* de Xilinx. La consommation de puissance totale est d'environ 0,549W, avec 0.527W pour la puissance statique et 0.022W pour la puissance dynamique. Notons que la différence de consommation de puissance du classificateur proposé est nettement significative comparé à un classificateur software.

Tableau 5.7 Puissance consommée pour différents circuits FPGAs

FPGAs	Total power (W)
Virtex 5	0.549
Virtex 6	1.295
Spartan3E	0.081 (static)
Virtex 7	0.546
Kintex 7	0.402

5.3 Evaluation des performances

5.3.1 Taux de reconnaissance

Le taux de reconnaissance maximal de notre architecture est d'environ 65,5%; ce taux est acceptable et comparable à celui obtenu avec des implémentations softwares. Le tableau 5.8 présente une comparaison du taux de classification du système proposé pour la classification des chromosomes implémenté sur FPGA avec les travaux existants (implémentations softwares) présentés dans le chapitre 1. Nous avons commencé par une évaluation en termes de taux de classification parce que ce facteur est la principale performance évaluée dans les classificateurs de chromosomes.

Le taux obtenu est similaire au taux de l'ANN probabiliste [38] pour des images de cellules femelles, et meilleur que le taux rapporté dans cette étude pour les images de cellules mâles. Il est également légèrement meilleur que le taux du classificateur décrit dans [11]. Cependant, il est inférieur à celui rapporté dans [14]; néanmoins, le résultat obtenu est acceptable.

Tableau 5.8 comparaison des performances en termes de taux de classification

Références	Méthodes (modèle ANN)	Taux de reconnaissance (%)
[14]	MLP	Probabilité de classification correcte des ensembles de tests 98%
[30]	ANN hybride MLP et Kohonen	Taux d'erreur maximum 33.1%
[39]	ANN Probabiliste	68.18% pour des images de cellules femelles 61.30% pour des images de cellules mâles
[11]	Multi-réseau de type Kohonen (implémentation software)	65.21%
Classificateur proposé	Multi-réseau de type Kohonen (implémentation hardware)	65.5%

Cette comparaison est approximative en raison de la différence entre les ensembles de données sur les chromosomes et les images utilisées dans chaque étude. De plus, ce taux de reconnaissance peut être amélioré en augmentant la qualité des images et la taille de la base de données utilisée.

Toutefois, l'objectif principal de ce travail, à savoir, la validation de la fonctionnalité du système de classification des chromosomes implémenté sur support hardware de type FPGA, a été atteint. L'amélioration principale avec la mise en œuvre hardware du classificateur de chromosomes est dans le temps d'apprentissage, c.-à-d. dans l'amélioration du temps de convergence comme souligné précédemment.

5.3.2 Evaluation des performances en termes de MCPS et MCUPS

Une évaluation des performances de l'architecture proposée en termes de MCPS et de MCUPS a été effectuée. Ces deux métriques sont communément utilisées dans l'évaluation des performances des implémentations hardware des différents modèles d'ANNs [81-83]. Nous exposerons dans cette section les travaux relatifs aux implémentations hardware du ANN de type Kohonen [75, 84-87]. Ce dernier étant le modèle choisi pour notre classificateur.

Le tableau 5.9 présente une comparaison de l'architecture proposée avec quelques travaux rapportés dans la littérature qui ne concernent pas forcément la classification de chromosomes (nous rappelons que nous proposons à travers ce travail de thèse, le premier modèle de classificateur hardware de chromosomes). La conception proposée atteint une performance de 1776 MCPS et de 1873 MCUPS pour des fréquences de 72,48MHz et 67.09MHz, respectivement.

Le résultat obtenu en terme de MCUPS est comparable à celui rapporté dans [85] et meilleur que ceux rapportés dans [84] et [87]. Dans [85] Henry *et al.* ont implémenté l'algorithme SOM sous forme d'un noyau IP en technologie ASIC 0.65µm CMOS *standard cells* avec une architecture SIMD (Single Instruction Multiple Data). Un FPGA a été utilisé pour implémenter le contrôleur externe. Les performances rapportées du core SOM-IP sont d'environ 1300 MCPS dans la phase de classification et de 660 dans la phase d'apprentissage, à une fréquence de 50MHz.

Dans [84], les auteurs ont proposé un accélérateur matériel pour le SOM, implémenté en 0.8µm CMOS AMS (*Austria Mikro System*) *standard cell*. Le SOM a été utilisé pour

l'analyse du processus de fabrication de circuits intégrés (ICs). Les auteurs ont rapporté des performances de 576 MCPS dans la phase de classification et de 48 MCUPS dans la phase d'apprentissage à 45 MHz. Les mêmes auteurs ont proposé un accélérateur matériel reconfigurable dynamiquement basé sur circuit FPGA (5FPGAs) pour la simulation du SOM [86]; ils ont rapporté une performance maximale de 17500 MCPS dans la phase de classification et 2000 MCUPS dans la phase d'apprentissage. Il est à noter aussi, que le MCPS obtenu pour notre architecture est significativement meilleur que le MCPS de l'architecture décrite dans [75].

Dans le travail exposé dans [87] Pena *et al.* ont proposé une architecture pour la carte de Kohonen unidimensionnelle et bidimensionnelle avec une fonction de voisinage exponentielle, ils ont exploré à la fois les métriques *Cityblock* (distance de Manhattan) et *Chessboard*, pour le calcul de la distance entre les vecteurs d'entrée et les vecteurs de poids. Ils ont rapporté une performance de 28,38 MCUPS pour la carte unidimensionnelle (en utilisant la *Cityblock*) implémentée sur un FPGA SpartanII xc2s400. Selon les auteurs, la *Cityblock* est plus appropriée en termes de surface utilisée et de vitesse, comparée à la métrique *Chessboard* dans le cas de la carte unidimensionnelle. Pour la carte bidimensionnelle, l'architecture basée *Cityblock* atteint une meilleure vitesse mais nécessite plus de surface que l'architecture basée sur la métrique *Chessboard*.

Tableau 5.9 Comparaison des performances en termes de MCPS et MCUPS

Références	Technologie	MCPS	MCUPS	Fréquence (MHz)
[75]	FPGA	9.78	4.89	200.2
[84]	ASIC (0.8 μ m)	576	48	45
[85]	ASIC (0.65 μ m)	1300	660	50
[86]	FPGA Virtex6	17500	2000	65
[87]	FPGA Spartan II	Non spécifié	28.38	Non spécifié
Architecture proposée	FPGA Virtex5	1776	1873	72,48 phase de classification 67.09 phase d'apprentissage

L'architecture proposée a été synthétisée sur différentes plateformes FPGA, pour évaluer les performances en termes de MCPS. Les résultats résumés dans le tableau 5.10 montrent que les MCPS obtenues sont comparables; la différence n'est pas vraiment significative. Ce résultat est dû au fait que les fréquences sont en quelque sorte similaires. Cependant, les résultats en terme de *maximum combinational path delay* est plus élevé pour les plateformes Kintex-7 KC 705 et Virtex7 VC 707, avec 5.01ns et 4.98ns pour la kintex7 et la Virtex7, respectivement. Le *maximum combinational path delay* est d'environ 3,48ns pour la plateforme utilisée dans notre implémentation, à savoir la ML501 (Virtex5).

Tableau 5.10 Comparaison des performances en termes de MCPS sur divers plateformes FPGAs de Xilinx

Plateforme FPGA	Circuit FPGA	MCPS	Max comb Path Delay (ns)
Plateforme Kintex-7 KC 705	XC7K325T-2FFG900	1248	5.01
Plateforme Virtex6 ML605	XC6VLX240T-1FFG1156	1940	3.21
Plateforme Virtex-7 VC707	XC7VX485T-2FFG1761	1252	4.98
Plateforme Virex5 ML501	XC5V50-3FFG676	1776	3.48

5.4 Conclusion

Des résultats prometteurs ont été obtenus, compte tenu de la partie réalisée du SoC, qui concerne le sous-système classification basé sur le réseau de neurones de type Kohonen. Ce sous-système est la partie principale du SoC, pour un système de classification des chromosomes humains, implémenté sur une plateforme FPGA.

Les résultats montrent l'efficacité du classificateur de chromosomes conçu sur un support hardware. Les résultats de synthèse montrent que l'architecture peut être implémentée sur une plateforme FPGA de type ML501-Virtex5. La partie réalisée du SoC-AKS utilise environ 14% de ressources. La consommation de puissance totale est estimée à 0.549W ; notons que la consommation de puissance pour le classificateur proposé est faible, vu que la solution est embarquée.

Le taux de classification de notre architecture est de 65,5%, ce taux est acceptable et comparable à celui obtenu avec des implémentations softwares. Nous soulignons que notre premier objectif était la validation du fonctionnement du système proposé pour la classification des chromosomes implémenté sur FPGA. Nous projetons l'utilisation d'une base de données plus riche avec des images de meilleure qualité dans le but d'améliorer ce taux de reconnaissance.

En termes de MCPS et de MCUPS, l'architecture proposée atteint une performance de 1776 MCPS et de 1873 MCUPS pour des fréquences de 72,48MHz et 67.09MHz, respectivement, ce qui est largement acceptable car nous ne recherchons pas un système qui fonctionne en temps réel mais plutôt un système rapide et portable.

Conclusions et perspectives

Ce travail constitue une contribution à la réalisation d'un SoC-AKS. Notre contribution consiste, principalement en la méthodologie proposée dans le but d'envisager une approche permettant l'implémentation d'un AKS sur support hardware. Notre développement est axé sur une implémentation typiquement hardware, dont l'intérêt est de permettre une amélioration des performances du système en exploitant le parallélisme inhérent aux ANNs via une implémentation sur une architectures reconfigurable de type FPGA.

Au cours de ce travail, nous avons établi un premier modèle architectural modulaire pour le SoC-AKS, pour garantir, d'une part, l'exécution parallèle, la portabilité, faible consommation de puissance et la miniaturisation d'un tel système, et d'autre part, obtenir de meilleures performances de fonctionnement en réduisant le temps de calcul lié à l'exécution software sur PC.

Des résultats prometteurs ont été obtenus, compte tenu de la partie réalisée du SoC, qui concerne le sous-système de classification basé sur le réseau de neurones de type Kohonen. Ce sous-système implémenté et testé avec succès sur une plateforme FPGA, est la partie principale du SoC-AKS. Les résultats montrent l'efficacité du sous-système conçu pour la classification des chromosomes sur un support hardware. Les résultats de synthèse montrent que l'architecture peut être implémentée sur une plateforme FPGA de type ML501-Virtex5. La partie réalisée du SoC-AKS utilise environ 14% de ressources. La consommation de puissance totale pour le classificateur proposé est estimée à 0.549W, cette dernière est faible, vu que la solution est embarquée.

Le taux de classification de notre architecture est de 65,5%, ce taux est acceptable et comparable à celui obtenu avec des implémentations logiciels. Nous soulignons que notre premier objectif était de valider le fonctionnement du système proposé pour la classification des chromosomes implémenté sur FPGA.

Nous avons proposé une méthodologie permettant l'implémentation d'un AKS sur support hardware de type FPGA. Bien que les résultats obtenus, compte tenu de la partie réalisée du SoC, sont satisfaisants, plusieurs améliorations et extension de ce travail peuvent être apportées. En effet, plusieurs perspectives sont envisagées, tout d'abord, à court terme, nous préconisons, comme première perspective, l'utilisation d'une base de données plus importante avec des images de meilleure qualité dans le but d'améliorer le taux de classification obtenu.

Le modèle architectural que nous proposons présente l'ossature pour la réalisation d'un SoC-AKS, facilitant ainsi la tâche pour d'éventuelle amélioration de ce travail. Une des extensions directes de notre travail est la poursuite des travaux sur l'intégration des autres parties du SoC, y compris le sous-système extraction des paramètres et le sous-système traitement des images des chromosomes, avec la même approche d'implémentation hardware. Il est important de prendre en considération les autres sous-systèmes afin de construire un système de classification automatique des chromosomes humains, et d'établir le caryotype et le diagnostic approprié avec une intervention humaine minimale.

Notre objectif à moyen terme est d'explorer d'autres approches de classification en hardware autres que l'approche ANNs, comme par exemple le SVM qui est utilisé dans des travaux récents et qui donne des taux de classification comparables à ceux de l'approche ANNs. De plus, il serait intéressant d'explorer d'autres approches de conception tel que le Co-design ou la conception haut niveau (High Level Synthesis (HLS)).

Le modèle architectural proposé et prototypé sur une plateforme FPGA est indépendant de la technologie, et peut être implémenté sur une autre technologie telle que la technologie ASIC (essentiellement celle basée sur les *standard cells*). Dans ce contexte, notre objectif à long terme, est de poursuivre le travail de recherche abordé

dans cette thèse, par l'adaptation de l'architecture proposée pour une implémentation sur support hardware de type ASIC pour de meilleures performances en termes de consommation de puissance et vitesse de calcul. En adoptant une approche hybride qui combine les deux supports, où un FPGA est utilisé pour le prototypage du design qui sera ciblé pour un ASIC après validation fonctionnelle. Ce concept de passage du FPGA vers l'ASIC est rendu possible avec la réutilisation d'IPs décrits au niveau RTL et peut s'avérer intéressant pour une réduction de temps de conception.

Bibliographies

- [1] J.H. Tjio and A. Levan, "The chromosome number of man", *Hereditas*, vol.42, no.1-2, pp. 1-6, May 1956.
- [2] A.T. Natarajan, "Chromosome aberrations: past, present and future", *Mutation Research*, vol. 504, no.1-2, pp. 3-16, Jul. 2002.
- [3] B. Lerner, "Towards a completely automatic neural-network-based and human chromosome analysis", *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 4, pp. 544-552, Aug. 1998.
- [4] M. Moradi, S.K. Setarehdan "New features for automatic classification of human chromosomes: A feasibility study", in *ELSEVIER Pattern Recognition Letters* 27; 2005, pp. 19-28.
- [5] B. Lerner and N.D. Lawrence, "A comparison of State-of-the-art classification Techniques with application to cytogenetics", *Neural Computing & Application*, vol. 10, no. 1, pp.39-47, Apr. 2001.
- [6] M. Popescu *et al.*, "Automatic karyotyping chromosomes of metaphase cells with overlapping", *Computers in Biology and Medicine*, vol. 29, no.1, pp. 61-82, Jan. 1999.
- [7] E. Poletti, A. Rugger, and E. Grisan, "Automatic classification of chromosomes in Q-band images", in *Proc.IEEE 30rd Int. Conf. Engineering in Medicine and Biology Society(EMBS)*, Vancouver, BC, Canada, 2008, pp. 1911-1914.
- [8] A. Carothers and J. Piper, "Computer-aided classification of human chromosomes, a review ", *Statistics and Computing*, vol. 4, no. 3, pp.161-171, Sept. 1994.
- [9] X. Wang, B. Zheng, M. Wood, S. Li, W. Chen, and H. Liu," Development and evaluation of automated systems for detection and classification of banded chromosomes: current status and future perspective", *Journal of Physics D: Applied Physics*, vol. 38, no. 15, pp. 2536-2542, Jul. 2005.
- [10] F. Abid and L. Hamami, "A survey of neural network based automated systems for human chromosome classification", *Artificial Intelligence Review*, vol. 49, no.1, pp. 41-56, Jan. 2018. DOI: 10.1007/s10462-016-9515-5.

- [11] A. Rahmania et L.Hamami, "Système de détection d'aberrations chromosomiques structurales", PFE, département d'informatique, Université des Sciences et de la Technologie Houari Boumediene (USTHB), Alger, Algérie, 2002..
- [12] H. BENHIDOUR, "Système de classification des chromosomes par réseaux de neurones", PFE, Institut National d'Informatique INI, Alger, Algérie, Sept.2001.
- [13] L. Hamami, H. Ait Abdesslam, R. HAROUN, "Chromosomal Automatic Classification System", in *Proc.17th IMACS World Congress*, Paris, France, Jul.2005.
- [14] B. Lerner, H. Guterman, I. Dinstein, and Y. Romem "Medial axis transform-based features and a neural network for human chromosome classification", *Pattern Recognition*, vol. 28, no. 11, pp. 1673-1683, Nov. 1995.
- [15] M. Moradi, S.K. Setarehdan S.R Ghaffari, "Automatic landmark detection on chromosomes' images for feature extraction purposes", in *Proc.3rd Int. Symp. Image and Signal Processing and Analysis*, Rome, Italy,2003.
- [16] Z. Kou, L. Ji, and X. Zhang, "Karyotyping of Comparative Genomic Hybridization Human Metaphases by Using Support Vector Machines", *Cytometry*, vol. 47, no.1, pp. 17-23, Jan. 2002.
- [17] C. Markou, C. Maramis, A. Delopoulos, C. Daiou, and A. Lambropoulos, "Automatic Chromosome Classification using Support Vector Machines", in *Pattern Recognition: Methods and Applications*, iConceptPress, Hong Kong, 2012.
- [18] A.S.Arachchige, J.Samarabandu, J.H.M.Knoll, and P.K Rogan, "Intensity integrated Laplacian-based thickness measurement for detecting Human metaphase chromosome centromere location". *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 7, pp. 2005-2013, Feb. 2013.
- [19] N. Madian and K.B. Jayanthi, "Analysis of human chromosome classification using centromere position ", *Measurement*, vol. 47, pp. 287-295,Jan.2014.
- [20] X. Wang, B. Zheng, S. Li, J.J. Mulvihill, and H. Liu, "A rule-based computer scheme for centromere identification", *Computer Methods and Programs in Biomedicine*, vol. 89, no. 1, pp. 33-42, Jan. 2008.
- [21] M. Moradi, S. K. Setarehdan, and S.R. Ghaffari, "Automatic locating of the centromere on human chromosome pictures", in *Proc. IEEE 16th Symp. Computer-Based Medical Systems (CBMS)*, New York, USA, Jun. 2003.
- [22] E. Poletti, A. Ruggeri, and E. Grisan, "An Improved classification Scheme for chromosomes with missing Data", in *Proc. IEEE 33rd Int. Conf. Engineering in Medicine and Biology Society(EMBS)*, Massachusetts,USA,2011, pp. 5072-5075. 1
- [23] J. Gregor and E. Granum, "Finding chromosome centromere using band pattern information", *Computer in Biology and Medicine*, vol. 21, no. 1-2, pp. 55-67, 1991.

- [24] R. Uttamatin, A. Intarapanich, S. Kaewkamnerd, P. Yuvapoositanon, and S. Tongsima, "Band classification based on chromosome shapes", in *Proc. IEEE Int. Conf. Biomedical Engineering*, Amphur Muang, Krabi, Thailand, Oct. 2013.
- [25] J.F. Jenq and S. Sahni, "Serial and Parallel Algorithms for the Medial Axis Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 12, pp. 326-333, Apr. 1992.
- [26] M. R. Mohammadi, "Accurate localization of chromosome centromere based concave points", *Journal of Medical Signals & Sensors*, vol. 2, no.2, pp. 88-94, 2012
- [27] D. Ming and J. Tian, "Automatic Pattern Extraction and Classification for Chromosome Images", *Journal of Infrared, Millimeter, and Terahertz Waves*, vol. 31, no. 7, pp. 866-877, Jul. 2010.
- [28] S. Khan, A. DSouza, J. Sanches, and R. Ventura, "Geometric Correction of Deformed Chromosomes for Automatic Karyotyping" in *Proc. IEEE 34th Int. Conf. Engineering in Medicine and Biology Society (EMBS)*, San Diego, California, USA, 2012.
- [29] Y. XIA, "Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.2, no.10, pp.1076-1086, Oct. 1989.
- [30] S. Eskiizmirliler, A.M. Erkmen, F. Basaran, M.S. Beksaç and A. Nur Çakar, "A hybrid intelligent diagnostic system based on neural networks and image analysis technique in the field of automated cytogenetics", in *Proc. Int. Conf. Image Processing*, 1996, pp. 315-318.
- [31] J. Theerapattanakul, J. Plodpai, C. Pintavirooj, and P. Lertprasert, "Feature Extraction for Automatic Chromosome Analysis", in *Proc. Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2005, pp 399-402.
- [32] E. Loganathan, M.R. Anuja, and N. Madian "Analysis of human chromosome images for the identification of centromere position and length", in *Proc. Point-of-Care Healthcare Technologies (PHT)*, Bangalore, India, Jan. 2013, pp.314-317.
- [33] X. Wang, B. Zheng, S. Li, J.J. Mulvihill, X. Chen, H. Liu, "Automated Identification Of Abnormal Metaphase Chromosome Cells For The Detection Of Chronic Myeloid Leukemia Using Microscopic Images", *Journal of Biomedical Optics*, vol. 15, no. 4, pp. 1-12, 2010 .
- [34] C.T. Falk, J.M. Gilchrist, M.A. Pericak-Vance, and M.C. Speer. "Using neural networks as an aid in the determination of disease status: Comparison of clinical diagnosis to neural-network predictions in a pedigree with autosomal dominant limb-girdle muscular dystrophy", *The American Journal of Human Genetics*, vol. 42, no. 4, pp. 941-949, Apr. 1998.

- [35] X. Wang, B. Zheng, S. Li, J.J. Mulvihill, M.C. Wood, H. Liu, "Automated classification of metaphase chromosomes: Optimization of an adaptive computerized scheme", *Journal of Biomedical Informatics*, vol. 42, no.1, pp. 22-31, Feb. 2009.
- [36] J. Graham, P.A. Errington, and A. Jennings, "A neural network chromosome classifier", *Journal of Radiation Research*, vol. 33, pp. 250-257, 1992.
- [37] P.A. Errington, and J. Graham, "Classification of chromosomes using a combination of neural networks", in *Proc. IEEE Int. Conf. Neural Networks*, San Francisco, CA, USA, 1993, pp. 1236-1241.
- [38] W.P. Sweeney, M.T. Musavi, and J.N. Guidi, "Probabilistic neural network as chromosome classifier", in *Proc. Int. Joint Conf. Neural Networks*, Nagoya, Japan, Oct. 1993, pp. 935-938.
- [39] S. Rungruangbaiyokand, P. Phukpattaranont, "Chromosome image classification using a two-step probabilistic neural network", *Journal of science and technologies*, vol. 32, no. 3, pp. 255-262, Jul. 2010.
- [40] B.C. Oskouei and J. Shanbehzadeh, "Chromosome Classification Based on Wavelet Neural Network", in *Proc. Int. Conf. Digital Image Computing: Techniques and Applications (DICTA)*, Sydney, NSW, Australia, Dec. 2010, pp. 605-610.
- [41] X. Ruan, 'A classifier with the fuzzy Hopfield network for human chromosomes', in *Proc. 3rd World Congress on Intelligent Control and Automation*, Hefei, China, Jul. 2000, pp. 1159-1164..
- [42] A.M. Badawi, K.G. Hasan, E.E.A. Aly, and R.A. Messiha, "Chromosomes classification based on neural network fuzzy rule based and template matching classifiers", in *Proc. 46th Midwest Symp. Circuits and Systems*, Cairo, Egypt, Dec. 2003, pp. 383-387.
- [43] E. Poletti, E. Grisan, A. Ruggeri, "A modular framework for the automatic classification of chromosomes in Q-band images", *Computer Methods and Programs in Biomedicine*, vol. 105, no. 2, pp. 120-130, 2012.
- [44] Bioimlab: <http://bioimlab.dei.unipd.it>
- [45] R. Uttamatani, P. Yuvapoositanon, A. Intarapanich, S. Kaewkamnerd, and S. Tongshima, "Chromosome Classification for Metaphase Selection", in *Proc. 13th Int. Symp. Communications and Information Technologies (ISCIT)*, Surat Thani, Thailand, Sept. 2013, pp. 464-468.
- [46] A. Narayanana, E.C. Keedwella, J. Gamalielsson, and S. Tatinenia, "Single-layer artificial neural networks for gene expression analysis", *Neurocomputing*, vol. 61, pp. 217-240, Oct. 2004.

- [47] V.N. Vapnik, "An overview of statistical learning theory", *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988-999, Sept. 1999.
- [48] B. Lerner, "Bayesian fluorescence in situ hybridisation signal classification", *Artificial Intelligence in Medicine*, vol. 30, no. 3, pp. 301-316, Mar. 2004.
- [49] H. Choi, K.R. Castleman, and A.C. Bovik, "Segmentation and fuzzy logic classification of M-FISH chromosomes images", in *Proc. IEEE Int. Conf. Image Processing*, Atlanta, GA, USA, Oct. 2006, pp. 69-72.
- [50] O. Sjahputera and J.M. Keller, "Evolution of a fuzzy rule-based system for automatic chromosome recognition", in *Proc. IEEE Int. Fuzzy Systems Conf.*, Seoul, South Korea, Aug. 1999, pp. 129-134.
- [51] M.P. Sampat, K.R. Castleman, and A.C. Bovik, "Pixel-by-pixel classification of MFISH images", in *Proc. 24th Conf. Engineering in Medicine and Biology Society and Fall Meeting of the Biomedical Engineering Society EMBS/BMES*, Houston, TX, USA, Oct. 2002, pp. 999-1000.
- [52] W.C. Schwartzkopf, A.C. Bovik, B.L. Evans, "Maximum-likelihood techniques for joint segmentation-classification of multispectral chromosome images", *IEEE Transactions on Medical Imaging*, vol. 24, no. 12, pp. 1593-610, Dec. 2005.
- [53] A. Lijiya, T.K. Mumthas, and V.K. Govindan, "Chromosome classification using M-FISH images", in *Proc. Int. Conf. Advances in Information Technology and Mobile Communication*, 2013, pp. 314-320.
- [54] Y.P. Wang, "M-fish image registration and classification", in *Proc. Int. Symp. Biomedical Imaging: Nano to Macro*, Arlington, VA, USA, Apr. 2004, pp. 57-60.
- [55] A.A. Jerraya, *Conception de haut niveau des systèmes monopuces*, Paris Hermès science publications Lavoisier, France, 2002.
- [56] R.A. Bergamaschi and J. Cohn, "The A to Z of SoCs", in *Proc. IEEE/ACM Int. Conf. Computer-aided design (ICCAD)*, San Jose, CA, USA, Nov. 2002, pp. 790-798.
- [57] C. Maxfield, *The Design Warrior's Guide to FPGAs, Devices, Tools and Flows*, 1st ed., Newnes, 2004.
- [58] <http://www.xilinx.com/>
- [59] www.altera.com/
- [60] <https://www.mentor.com/>
- [61] K.C. Wu and Y.W. Tsai, "Structured ASIC, evolution or revolution?", in *Proc. IEEE/ACM Int. Symp. Physical design (ISPD)*, Phoenix, Arizona, USA, Apr. 2004, pp. 103-106.
- [62] http://en.wikipedia.org/wiki/Structured_ASIC_platform

- [63] F. Nekoogar, F. Nekooga, *From ASICs to SoCs: A Practical Approach*, 1st ed., Prentice Hall PTR publisher, USA, 2003.
- [64] Neil H. E. Weste, David Money Harris, *CMOS VLSI Design a Circuits and Systems Perspective*, 4th ed., Harvey Mudd College Addison-Wesley, USA, 2011.
- [65] www.cadence.com/
- [66] www.Synopsys.com/
- [67] Design Compiler User Guide, Version F-2011.09-SP2, December (2011), Synopsys
- [68] Cadence Design Systems, GUI user guide for Encounter RTL Compiler, Product Version 14.10
- [69] P. Coussy, M. Meredith, D.D. Gajski, A. Takach, “An Introduction to High-Level Synthesis”, *IEEE Design & Test of Computers*, vol. 26, no. 4, pp. 8-17, 2009.
- [70] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers and Z. Zhang, “High-Level Synthesis for FPGAs: From Prototyping to Deployment”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 473-491, Apr. 2011.
- [71] Vivado Design Suite User Guide *High-Level Synthesis* UG902 (v2016.1) April 6, 2016
- [72] A. P. Britto and G. Ravindran, “A review of cytogenetics and its automation”, *Journal of Medical Sciences*, vol. 7, pp. 1-18, 2007.
- [74] D.D. Gajski and R.H. Kuhn, “New VLSI Tools”, *Computer*, vol. 16, no. 12, pp. 11-14, Dec. 1983. DOI:10.1109/MC.1983.1654264.
- [74] T. Kohonen, *Self-Organization and Associative Memory*, 2nd ed., Springer Series in Information Science, Springer-Verlag Berlin Heidelberg, Germany, 1987.
- [75] H. Hikawa, “FPGA implementation of self-organizing map with digital phase locked loops”, *Neural Networks*, vol. 18, no. 5-6, pp. 514–522, 2005.
- [76] A. Rajah, M.K. Hani, “ASIC Design of a Kohonen Neural Network Microchip”, in *Proc. IEEE Int. Conf. Semiconductor Electronics*, Kuala Lumpur, Malaysia, Dec. 2004, pp. 148-15.
- [77] F. Abid, L. Hamami, F. Badache, and H. Derdour, “Hardware implementation of Kohonen neural network for human chromosomes classification”, in *Proc. 2nd Int. Conf. Complex Systems (CISC)*, Jijel, Algeria, Dec. 2011.
- [78] **F. Abid**, L. Hamami, F. Badache and H. Derdour, “Design and implementation of Kohonen neural network based human chromosomes classification system on chip “

in *Proc. Journées d'Etudes Algéro-Françaises de doctorants en Signal-Image & Applications (JEAFD)*, Algiers, Algeria, Dec. 2012.

- [79] **F. Abid** and L. Hamami, “FPGA-based system for human chromosome Classification”, in *Proc. Engineering and Technology*, vol.22, Mar. 2017, pp.13-16.
- [80] **F. Abid**, L. Hamami, F. Badache, and H. Derdour, “A System on Chip for Automatic Karyotyping System”, *Computers and Electrical Engineering*, vol. 64, pp. 1-14, Nov. 2017. DOI: doi.org/10.1016/j.compeleceng.2017.10.001.
- [81] J. Misra and I. Saha, “Artificial neural networks in hardware: A survey of two decades of progress”, *Neurocomputing*, vol. 74, no. 1–3, pp. 239-255, Dec. 2010.
- [82] N. Izeboudjen, C. Larbes, and A. Farah,” A new classification approach for neural networks hardware: from standards chips to embedded systems on chip”, *Artificial Intelligence Review*, vol. 41, no. 4, pp. 491-534, Mar. 2012.
- [83] A. R. Omondi and J. C. Rajapakse, *FPGA Implementations of Neural Networks*, Springer-Verlag New York, US, 2006.
- [84] M. Pormann, U. Witkowski, H. Kalte, and U. Rückert, “Dynamically Reconfigurable Hardware – A New Perspective for Neural Network Implementations”, in *Proc. 12th Int. Conf. Field Programmable Logic Applications*, Montpellier, France, Sept.2002, pp. 1048-1057.
- [85] D.C. Hendry, A.A. Duncan, N. Lightowler, “IP Core Implementation of a Self-Organizing neural network”, *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1085 - 1096, Sept. 2003.
- [86] M. Pormann, U. Witkowski, U. Rückert, “A Massively Parallel Architecture for Self-Organizing Feature Maps”, *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1110-1121, Sept.2003.
- [87] J. Pena, M. Vanegas, A. Valencia, “Digital hardware architectures of Kohonen’s Self Organizing Feature Maps with exponential neighboring function”, in *Proc. IEEE Int. Conf. Reconfigurable Computing and FPGA's*, San Luis Potosi, Mexico, Sept. 2006.

ANNEXE A : Généralités sur les chromosomes

A.1 Introduction

La cytogénétique, qui est la science qui traite l'analyse des chromosomes, leur rôle dans la transmission du matériel génétique et leur classification. Pour analyser les chromosomes, les cytogénéticiens utilisent généralement, le sang compte tenu de sa simplicité de prélèvement et la rapidité de la culture de ses cellules; l'acide amniotique est aussi utilisé dans l'analyse des chromosomes. Ce n'est qu'en 1956 que le nombre de 46 chromosomes est attribué aux chromosomes des cellules humaines; ce nombre a été établi par *Tjio* et *Levan*. Les chromosomes sont organisés en 23 paires dont 22 paires d'autosomes et une paire de gonosomes ou chromosomes sexuels, XX chez la femme, XY chez l'homme.

Les 23 paires sont représentées après classification dans un format standard appelé caryotype. Le processus d'établissement du caryotype est une partie essentielle de la cytogénétique, car il permet le diagnostic des syndromes génétiques qui découlent des aberrations chromosomiques.

A.2 Les chromosomes

Les chromosomes sont des structures microscopiques présentes dans le noyau de la cellule vivante. Ces derniers, sont une succession de gènes porteurs du matériel héréditaire de la cellule parentale à la cellule fille. Ils sont constitués de molécule d'ADN (Acide Désoxyribo Nucléique) qui représente le support de l'information génétique. La cellule humaine normale comporte 46 chromosomes, qui sont visibles sous forme de bâtonnets au stade de la métaphase. La métaphase est une phase statique de la mitose (la division cellulaire qui donne naissance à deux cellules filles identiques à la cellule mère) où les chromosomes sont individualisés, vu qu'ils sont à leur condensation maximale.

A.2.1 Morphologie du chromosome

La figure A.1 représente la structure du chromosome métaphasique, qui est constitué de deux chromatides reliés par une constriction appelée centromère. Les chromatides sont terminés par les télomères (structure qui rend le chromosome stable, et évite la fusion de l'extrémité du chromosome avec d'autres chromosomes).

Le centromère divise le chromosome en deux bras : le bras court (petit "p") et bras long (queue "q"). Chaque chromosome des 23 paires est déterminé par sa taille (longueur), la position de son centromère et les bandes claires et sombres qui le caractérisent. Le rapport du bras court sur la longueur totale du chromosome définit l'indice centromérique (IC), qui constitue un des paramètres chromosomiques utilisé pour la classification.

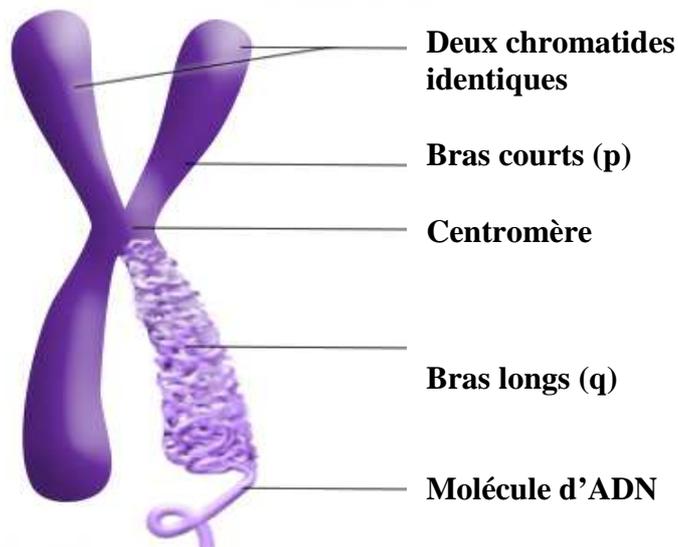


Figure A.1 Structure du chromosome

A.2.2 Le caryotype

Le caryotype humain (Figure A.2-b-) est la représentation standard des 46 chromosomes d'une cellule en métaphase. Les 46 chromosomes sont classés en paires de 22 autosomes et 1 paire de chromosomes sexuels. Le caryotype permet d'établir le nombre et la morphologie des chromosomes et de les classer par taille décroissante et par position du centromère. Le caryotype est la méthode d'analyse du matériel chromosomique qui permet de détecter d'éventuelles réorganisations des chromosomes et de déduire les anomalies chromosomiques qui en découlent.

Avant les années 70, les méthodes cytologiques étaient basées sur les caractéristiques géométriques des chromosomes, à savoir leur taille et la position du centromère, cette restriction est due aux méthodes de coloration utilisées, ces dernières donnent une coloration uniforme aux chromosomes. Avec les méthodes classiques les chromosomes étaient considérés comme des structures homogènes. Cela permet de détecter les anomalies de nombre (gain ou défaut d'un chromosome) mais ces méthodes s'avèrent non pratiques pour détecter des anomalies de structure (perte d'un segment de chromosome par exemple). C'est ainsi que les techniques de marquage dites *banding* sont apparues. Ces méthodes utilisaient des colorants (*Giemsa*) et la fixation d'un fluorochrome (*quinacrine*), pour mettre en évidence des bandes tout au long du chromosome. Le chromosome apparaît sous forme d'une distribution de bandes claires et sombres suivant une séquence propre à chaque chromosome (paire de chromosomes).

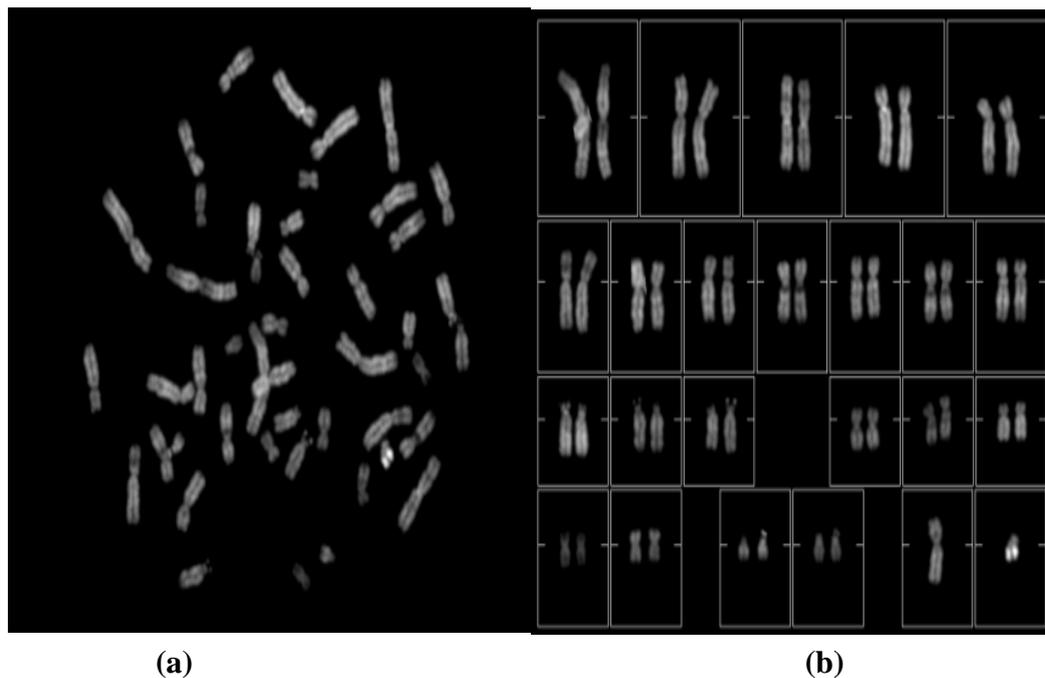


Figure A.2 La cellule métaphasique et son caryotype

(a) cellule métaphasique (b) Caryotype [14]

Avec ces méthodes l'établissement du caryotype se fait en se basant sur les paramètres géométriques du chromosome, et sur les paramètres structurels relatifs aux informations des bandes, en particulier la densité de profil. Ainsi l'analyse structurelle

est plus précise et des anomalies éventuelles de structure sont détectées. La carte standard de distribution des bandes (G) de chaque chromosome du caryotype humain, où on peut observer jusqu'à 850 bandes, est appelée caryogramme.

A.3 Techniques de marquage des chromosomes métaphasiques

Le marquage des chromosomes par des techniques appropriées est requis pour leur identification. Plusieurs méthodes de marquage ont été appliquées dans le but d'identifier les chromosomes d'une cellule. Par exemple, dans la technique de marquage dite *banding*, le chromosome est distingué par des bandes claires et sombres. Une nouvelle nomenclature internationale (Paris 1971) est ainsi définie pour répertorier ces bandes. Chaque région des quatre qui constituent le chromosome est divisée en bandes numérotées.

Exemple: 6p12 correspond à la deuxième bande de la région 1 du bras court du chromosome numéro 6. On distingue suivant la technique appliquée, les bandes chromosomiques suivantes:

A.3.1 Les bandes G (*Giemsa*)

Ce sont les bandes sombres qui apparaissent après marquage des chromosomes avec le *Giemsa* qui se lie à l'ADN, les bandes G plus ou moins claires sont dites bande G-négatives.

A.3.2 Les bandes R (*Reverse*)

Ces bandes apparaissent après traitement des chromosomes à 87° C dans une solution à PH 6.5 pendant quelques minutes. Le *Giemsa* fait apparaître des bandes R dont l'organisation est l'inverse des bandes G. Cette technique permet la détection de petites réorganisations de structure sur les extrémités des chromosomes (télomères), qui sont considérablement marquées par cette technique.

A.3.3 Les bandes Q (*Quinacrine*)

Ce sont des bandes qui apparaissent sur le chromosome après coloration par des dérivés de *quinacrine* appelées moutarde de *quinacrine*, qui fluorescent avec la

lumière ultraviolette. On distingue avec cette technique principalement, les chromosomes Y dont la partie terminale du bras long est intensément fluorescente.

A.3.4 Les bandes C (Centromère)

Ce sont les bandes correspondant au centromère qui apparaît sombre, en particulier pour les chromosomes 1, 9 et 16.

A.3.5 Technique de haute résolution

Mise à part la technique G-banding sur laquelle est basée la cytogénétique classique, d'autres techniques de marquage des chromosomes, qui s'ajoutent à la technique G-banding, sont apparues pour une cytogénétique moléculaire. Entre autre les techniques: Hybridation in situ fluorescente (FISH : *Fluorescence In-situ Hybridization*), Hybridation Génomique Comparative (CGH : *Comparative Genomic Hybridization*) et la technique multicolores hybridation in situ fluorescente ou M-FISH (*Multiplex-FISH*). Ces techniques sont employées dans les caryotypes en couleurs. La technique M-FISH utilise 5 colorants et un extra fluorophore, DAPI (*4-6-Diamidino-2-phenylindole*); avec cette technique chaque paire de chromosomes apparaît avec une couleur distincte, ce qui permet la détection des aberrations structurales. La CGH utilise trois colorants (rouge, vert et bleu), cette technique est dite globale car elle concerne le génome.

A.4 Les anomalies chromosomiques

L'analyse du caryotype humain permet de détecter les anomalies chromosomiques et les syndromes cliniques [2]. Ces aberrations sont le résultat d'un changement ou réorganisation du nombre des chromosomes ou de leurs structures. Ces changements sont les causes de plusieurs maladies génétiques ou maladies par aberrations chromosomiques. Ainsi, l'analyse de la morphologie du chromosome est une procédure importante pour le diagnostic des cancers et des maladies génétiques. Les anomalies chromosomiques ont été décrites entre 1959 et 1965, elles sont classées, principalement en deux catégories, à savoir les anomalies numériques et les anomalies structurales. Les anomalies les plus fréquentes sont résumées dans le tableau A.1.

A.4.1 Les anomalies numériques

La cellule normale est constituée d'un nombre fixe de paires de chromosomes (23 paires). Les anomalies numériques se présentent sous forme d'un surnombre (trisomie) ou un déficit de chromosome (monosomie). Dans le cas d'une trisomie, les chromosomes d'une paire se présentent en trois chromosomes au lieu de deux, cette anomalie affecte, généralement les paires suivantes : La paire 13 (*Patau syndrome*), la paire 21 (*Down syndrome*), la paire 18 (*Edward syndrome*) et les chromosomes sexuels (*Klinefelter syndrome* et *Turner syndrome*).

A.4.2 Les anomalies structurelles

Les anomalies structurelles (figure A.3) sont les changements qui affectent la structure du chromosome. On distingue plusieurs types, comme la **délétion** (perte d'un matériel génétique), l'**insertion** (présence d'un matériel génétique additionnel), l'**inversion** (cassure d'un segment de chromosome qui est tourné et rattaché par la suite, ce qui conduit à une inversion du matériel génétique). Les **translocations** (échange de matériel génétique entre deux chromosomes). Les translocations sont présentes en deux types, les translocations réciproques et Robertsonniennes (fusion de deux chromosomes par leurs centromères).

Les aberrations chromosomiques sont les causes d'environ 50% des avortements, de 10% des retards mentaux et la cause de plusieurs cancers [9].

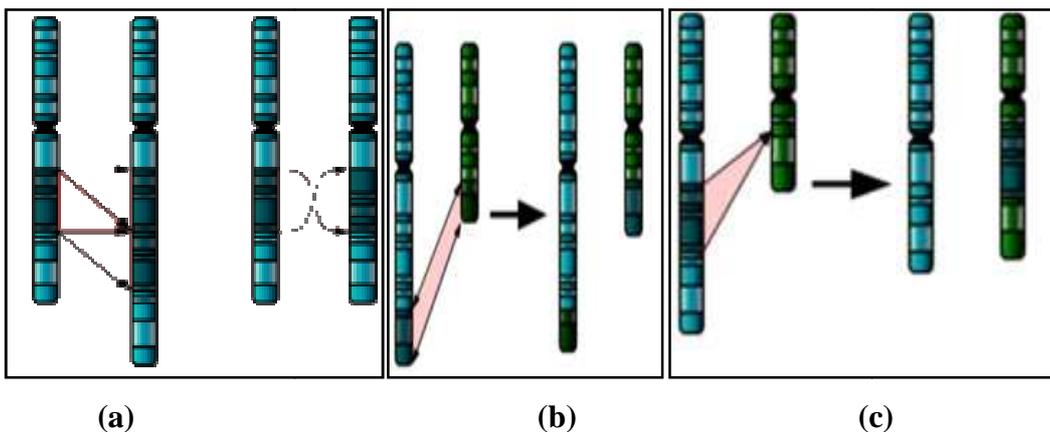


Figure A.3 Aberrations structurelles

(a) délétion et inversion (c) insertion (d) translocation

Tableau A.1 Les anomalies chromosomiques les plus fréquentes

Anomalies Chromosomiques			
Anomalies numériques		Anomalies structurelles	
Type d'anomalie	Description	Type d'anomalie	Description
Polyploidy	Triploïde : extra chromosomes, présence de 3 chromosomes dans chaque paire.	Délétions	Perte d'un matériel génétique (ex. délétion dans le bras p du chromosome 5, formule chromosomique, 46,XX,-5p).
	Tétraploïde : présence de 4 chromosomes dans chaque paire.		
Aneuploidy	Monosomie Défaut d'un chromosome, ex. : syndrome de <i>Turner</i> (formule chromosomique, 45, X0)	Insertions	Présence d'un matériel génétique additionnel.
		Inversions	Cassure d'un segment de chromosome qui est tourné et rattaché par la suite, ce qui conduit à une inversion du matériel génétique.
	Trisomie Extra chromosome dans des paires d'autosomes : paire 21 (syndrome de <i>Down</i> , formule chromosomique 47, XY, 21). paire 13 (syndrome de <i>Patau</i> , formule chromosomique 47, XY,13). paire 18 (syndrome d' <i>Edward</i> , formule chromosomique 47, XY,18). Extra chromosome dans la paire de gonosomes (syndrome de <i>Klinefelter</i> , formule chromosomique 47, XXY).	Translocations	Echange de matériel génétique entre deux chromosomes ex. translocation entre les chromosomes 1 et 18, formule chromosomique 46, XX, t(1;18) Les translocations sont présentes en deux types, les translocations réciproques et Robertsonniennes (fusion de deux chromosomes par leurs centromères).
		Isochromosomes	Chromosomes avec des bras identiques.

ANNEXE B : Algorithme de Kohonen

B.1 Réseau de Kohonen

Le réseau de Kohonen est un réseau à compétition dont la topologie est celle d'une carte auto-organisatrice ou SOM (*Self Organizing Map*). Il est constitué de deux couches; la couche d'entrée contient les vecteurs à classer et la couche de sortie contient des neurones qui rentrent en compétition dans la phase d'apprentissage de type non-supervisé. La liaison entre l'ensemble de I entrées et les J unités de sortie est réalisée par les poids W_{ij} . Les unités proches dans la carte (*map*) correspondent à des classes voisines.

B.2 Topologie et notion de voisinage

Dans la carte de Kohonen les neurones sont organisés, généralement dans un réseau unidimensionnel ou bidimensionnel. Le nombre de neurones dans le réseau et leur disposition caractérisent la topologie du réseau. La figure B.1 illustre un réseau bidimensionnel dans une topologie carrée; généralement, les neurones de la carte sont disposés dans un voisinage carré ou hexagonal. L'ensemble de J neurones localisés sur le réseau à une distance inférieure ou égale au rayon de voisinage (r) forme le voisinage ($V_r(j^*)$) du neurone gagnant (j^*), celui ayant la plus petite distance. La notion de voisinage entre les neurones fait la particularité de ce réseau.

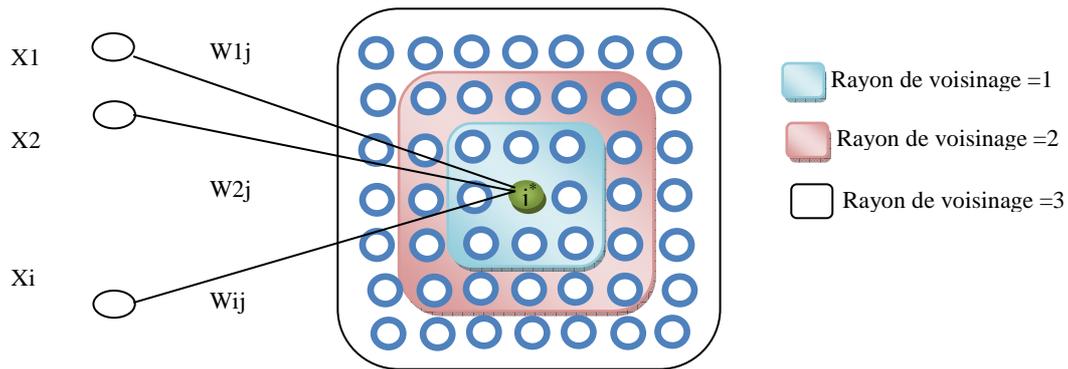


Figure B.1 Topologie de voisinage pour une carte de Kohonen carrée

B.3 Algorithme de Kohonen

L'algorithme de Kohonen développé en 1984 est un algorithme d'apprentissage non-supervisé qui met en correspondance l'espace des entrées de dimension I et la carte de dimension J . Cet algorithme consiste en trois processus, à savoir, l'initialisation des vecteurs de poids, le processus de compétition (détermination du neurone gagnant) et celui de l'adaptation (mise à jour des poids synaptiques). Les deux derniers processus (étapes) de l'algorithme de Kohonen sont illustrés par la figure B.2.

B.3.1 Initialisation des vecteurs de poids

La première étape consiste en l'initialisation aléatoire des vecteurs de poids. La dimension du vecteur de poids de chaque neurone dans le réseau est égale à la dimension du vecteur d'entrée. L'initialisation appropriée consiste à choisir aléatoirement de petites valeurs différentes de $W_j^0, j \in [1, J]$ où J est le nombre de neurones dans la carte.

B.3.2 Compétition

La compétition se fait dans un voisinage autour d'un neurone gagnant. Pour chaque vecteur d'entrée $X = (x_1, x_2, \dots, x_i)$, la distance Euclidienne à tous les vecteurs de poids de la carte est calculée afin de déterminer le neurone gagnant qui est le

neurone le plus proche, et donc celui avec la plus petite distance, cette distance est calculée selon la formule (B.1).

$$d_j^2 = \sum_{i=1}^I (x_i(t) - W_{ij}(t))^2 \quad (\text{B.1})$$

Le neurone gagnant (j^*) est déterminé selon le critère de distance minimale exprimé par (B.2).

$$j^* = \operatorname{argmin} \|x_i - w_j\| \quad (\text{B.2})$$

B.3.3 Adaptation

Après détermination du neurone gagnant, son vecteur de poids et les vecteurs poids des neurones dans son voisinage sont ajustés au vecteur d'entrée pour être organisés en régions. L'adaptation des poids W_{ij} est faite de telle manière que des exemples proches dans l'espace d'entrée soient associés au même neurone ou à des neurones proches dans la carte.

Le processus de mise à jour est une variation de l'emplacement des neurones, c'est-à-dire que si le neurone est à l'intérieur du voisinage, ses poids seront ajustés, sinon sa position reste inchangée. La mise à jour des poids se fait selon la formule (B.3).

$$W_{ij}^{n+1} = W_{ij}^n + \varepsilon_n h_n^{j,j^*} (x^n - W_{ij}^n) \quad (\text{B.3})$$

$$i \in [1, I], j \in [1, J]$$

$$h_n^{j,j^*} = \exp\left(-\frac{\|j - j^*\|^2}{2\sigma_n^2}\right) \quad (\text{B.4})$$

Où ε_n est le taux d'apprentissage, c'est un facteur qui contrôle la variation des poids à l'itération n .

h_n^{j,j^*} est la fonction de voisinage, exprimée par (B.4). Elle définit l'influence du neurone gagnant (j^*) sur son neurone voisin (j). La fonction de voisinage diminue avec la distance entre la position du neurone gagnant (j^*) et la position du neurone voisin (j) dans la carte.

σ : est le coefficient de voisinage; la décroissance de la taille du voisinage s'obtient par diminution de ce paramètre.

1- Les valeurs de $h_n^{jj^*}$ sont dans l'intervalle $[0,1]$, elles valent 1 lorsque $j=j^*$ (le neurone considéré et le neurone gagnant) et décroissent quand la distance entre le neurone gagnant et le neurone considéré augmente.

2- Les valeurs des deux paramètres σ et ε sont réduites graduellement au cours de l'apprentissage. En effet, si la valeur de σ est grande plusieurs neurones se rapprochent de l'entrée X, et dans le cas contraire (σ est petit), l'adaptation reste très localisée. D'autre part, si la valeur de ε est grande il y aura un risque d'instabilité, et dans le cas contraire, (ε petit) l'adaptation est faible.

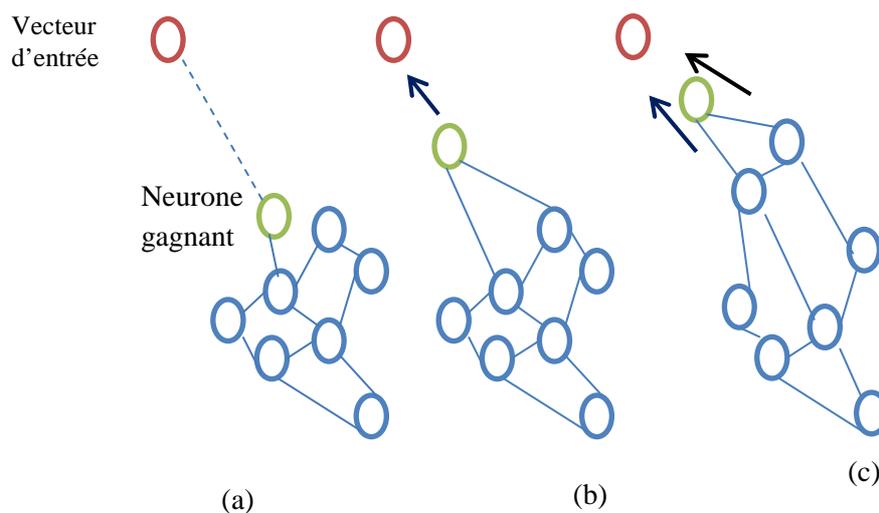


Figure B.2 Illustration de l'algorithme de Kohonen

(a) Sélection du neurone gagnant (compétition)

(b) Déplacement du neurone gagnant vers le vecteur d'entrée

(c) Déplacement des neurones voisins du neurone gagnant vers le vecteur d'entrée (adaptation)

A la fin de l'apprentissage on obtient une mémoire contenant l'ensemble des vecteurs poids associés aux neurones J de la carte. Ces vecteurs de poids mis à jour sont utilisés

dans la phase de classification, où des vecteurs d'entrée sont présentés au réseau et on détermine le neurone le plus proche de l'entrée en terme de distance. Dans le cas d'un problème de classification, le neurone le plus proche représente la classe du vecteur d'entrée.