REPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA DE BOUMERDES

FACULTE DES SCIENCES
DÉPARTEMENT DE PHYSIQUE
STRUCTURE INFOTRONIQUE

Science et Techniques
Filière : Génie Electrique
Spécialité : Systèmes Electroniques complexes

Mémoire de fin d'études présenté en vue D'obtention du diplôme de Master
En
Systèmes Electroniques Complexes

Thème :

IMPLEMENTATION DES ONDELETTES GEOMETRIQUES POUR LA SEGMENTATION D'IMAGES MEDICALES

Réalisé par :

BENFATTOUM Nora
BOUDJENAH Ibrahim

Promoteurs :

Mr B. HACHEMI (CDTA)
Mr F.TALBI (CDTA)
Mme H.BOUMERIDJA (UMBB)

Centre d'accueil : Centre de Développement et de Technologies Avancées d'Alger
Année universitaire : 2015-2016

Dédicace

Je dédie ce modeste travail à tous ceux qui me sont chers:

***A** l'être le plus cher pour moi dans cette vie, ma mère et,*

À l'être que je respecte le plus dans ce monde, mon cher père,

pour leur sacrifice et encouragement pour finir ce travail. Je leur souhaite une vie plein de joie, de santé et de bonheur.

A ma tante, ma soeur et ma meilleure amie Khadidja

A mes chers frères Abdennour, Said, Abderahmane et la petite Taïma.

Dieu me les garde pour le reste de ma vie

Et Surtout à mon frère Ammar qui j'ai récemment perdu « Que Dieu ait son âme ».

***A** ma grande Famille*

***A** mes chères amies*

Nora

Dédicace

Je dédie ce modeste travail à tous ceux qui me sont chers:

A l'être le plus cher pour moi dans cette vie, ma mère et,

À l'être que je respecte le plus dans ce monde, mon cher père,

pour leur sacrifice et encouragement pour finir ce travail. Je les souhaite une vie pleine de joie, de santé et de bonheur.

A mes chères frères Islam et Mohamed Amin

Et mes chères sœurs ainsi que mes nièces et neveux Yasmine, Yasser, Sohaib, Aïcha

Dieu me les garde pour le reste de ma vie

A ma grande Famille

A mes chers amis Zaki, Hachim

Ibrahim

Remerciements

*Au Tout Puissant **ALLAH** pour nous avoir donné la santé et la force de mener ce travail à bien.*

A l'issue de ce travail, nous voudrions adresser notre profonde reconnaissance et nos sincères remerciements à :

*Mr.BELKACEM et Mr.TALBI de nous avoir encadré au sein de l'équipe de recherche, AC2 (Architectures pour la classification et cryptographie), de la division ASM et à leur tête **M^{me} D.ALIM**, Chef d'équipe, pour nous avoir accueilli et de nous avoir permis de travailler dans de bonnes conditions.*

. Merci à vous d'avoir accepté de diriger ce travail, merci pour votre perfectionnisme, soutien, disponibilité, et gentillesse, merci pour votre sympathie et vos encouragements et pour avoir partagé vos compétences avec nous tout au long de ces mois ainsi pour l'inspiration, l'aide et le temps que vous avez bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour, Que vous trouvez ici l'expression de notre plus profond respect et nos sincères gratitudes.

Nous tenons à remercier notre promotrice, Mme H. BOUMERIDJA CHEBAANI pour la qualité de son enseignement, pour nous avoir guidés dans ce mémoire.

Nos remerciements vont aux membres du jury pour l'honneur qu'ils nous ont fait de participer à la soutenance de ce projet de fin d'études.

Nous n'oublions pas l'ensemble du personnel du CDTA qui a participé au bon déroulement de notre stage pratique.

Un très grand merci à tout le corps enseignant de la structure d'Infotronique de l'université M'hamed BOUGARA de Boumerdes qui nous a guidés durant toutes nos années d'étude.

Résumé :

L'objectif principal de ce travail est d'implémenter sous l'environnement Vivado 2015.4 sur un circuit FPGA (Field Programmable Gate Array) une ondelette géométrique pour la détection de contour.

Pour cela nous avons étudié la transformée en bandelettes en se basant sur l'algorithme développé par *Le Pennec* et *Peyré*, ensuite nous avons remplacé la transformée d'ondelette utilisée dans les bandelettes par une transformée d'ondelette basée sur l'approche du *lifting schème* dan le but est de l'implémenter sur un circuit FPGA (ZedBoard).

Abstract:

The main objective of this work is to implement in the Vivado 2015.4 environment on an FPGA circuit (Field Programmable Gate Array) a geometric wavelet for edge detection. For this we have studied bandlets transform developed by *Le Pennec* and *Peyre*, and then we replaced the wavelets algorithm in the bandlets with *lifting scheme* wavelets approach to be implemented on the FPGA platform.

ملخص

الهدف الرئيسي من هذا العمل هو زرع في ظل بيئةVivado2015.4 على الدارة المبرمجة FPGA المويجات الهندسية للكشف عن حواف الصور

لهذا قمنا بدراسة خوارزمية التحويل إلى شرائط حيث ركزنا على الخوارزمية المطورة من طرف بيير وبنك وبعد دلك استبدلنا خوارزمية تحويل المويجات المطبقة في خوارزمية الشرائط إلى خوارزمية تحويل المويجات عن طريق مخطط الرفع لزرعها في الدارة المبرمجة FPGA(ZedBoard)

Table des matières

Table des figures

N° Figure Titre Page

Liste des tableaux

Table des abréviations

Abréviation	*Sens*
2D	2 Dimensions
ASIC	***A**pplication **S**pecific **I**ntegrated **C**ircuit*
CLB	***C**onfigurable **L**ogic **B**lock*
CPLD	***C**omplex **P**rogrammable **L**ogic **D**evice*
DWT	**D**iscrete **W**avelet **T**ransforms
EPLD	***E**rasable **P**rogrammable **L**ogic **D**evice*
FPGA	***F**ield **P**rogrammable **G**ate **A**rrays*
HDL	***H**igh speed integrated circuits **H**ardware **D**escription **L**anguage*
IC	***I**ntegrated **C**ircuit*
IRM	***I**magerie par **R**ésonance **M**agnétique*
ISE	***I**ntegrated **S**ynthesis **E**nvironment*
LB	***L**ogic **B**locks*
LUT	***L**ook **U**p **T**able*
MUX	***Mu**ltip le**x**er*
PL	***P**rogrammable **L**ogic*
PLD	***P**rogrammable **L**ogic **D**evice*
PS	***P**rocessing **S**ystem*
RAM	***R**andom **A**ccess **M**emory*
SDK	***S**oftware **D**evelopment **K**it*
SOC	**S**ystem **O**n **C**hip
TOC	**T**ransformée en **O**ndelettes **C**ontinues
TOD	**T**ransformée en **O**ndelettes **D**iscrètes
VHDL	***V**ery **H**igh speed integrated circuits hardware **D**escription **L**anguage*

Introduction générale

Plusieurs outils mathématiques ou transformées efficaces pour le traitement d'image médicale (segmentation, classification, compression, le débruitage….) existent, ils sont en général des combinaisons d'un ensemble de transformées avec plus au moins une adaptation pour un certain type de données ou d'applications.

Chacune de ces transformées permet de réaliser un meilleur compromis selon un ou plusieurs axes. Ces transformées sont plus au moins complexes et coûteuses en temps de calcul avec une variation de la qualité selon la complexité de ces transformées. Pour les différents transformées le problème le plus difficile est la détermination de la complexité de ces transformées qui dépend en général de l'application, qui peut être le nombre de cycle, la quantité de la mémoire, la bande passante de la mémoire, le nombre de portes logiques. Pour pouvoir comparer ces différentes transformées, nous utilisons généralement une mesure de performance d'une implémentation matérielle.

Ces transformées présentent de nouveaux défis pour être implémentées et optimisées. Vu les possibilités offertes par ces transformées , elles sont potentiellement candidate pour être adoptée pour un grand nombre d'applications telles que l'imagerie médicale, le cinéma numérique, la surveillance, les satellites, les applications mobiles (appareil photo, PDA, téléphone cellulaire,…). Ces différentes applications présentent plusieurs exigences et contraintes : le temps, la qualité, ou la consommation d'énergie,…. Vu cette variété d'applications et d'exigences, trouver une implémentation et une architecture pouvant satisfaire l'ensemble de ces contraintes se révèlent très difficile. Quelle implémentation et quelle architecture choisir ?

Parmi ces transformées citons la DWT (**D**iscrete **W**avelet **T**ransforms) qui est un outil mathématique d'un grand intérêt pour la segmentation et la détection de contour; particulièrement efficace pour différentes applications telles que l'imagerie médicale.

L'imagerie médicale a profondément influencé à la fois la recherche médicale et la pratique clinique. Elle est devenue incontournable aussi bien pour l'établissement d'un diagnostic que pour la mise en place et le suivi d'un traitement thérapeutique. Elle fournit un volume croissant de données tridimensionnelles provenant de modalités d'acquisitions différentes. Nous citons le scanner-X, la médecine nucléaire, l'échographie, l'Imagerie par résonance magnétique IRM. Ce volume important d'informations rend délicates et laborieuses les tâches d'analyse et d'interprétation par un expert.

L'interprétation du diagnostic des images médicales s'effectue en plusieurs étapes dans le but de détecter des anomalies. Ce but est précisément atteint lorsque le clinicien intègre les deux procédés suivants. Le premier est la perception de l'image pour reconnaitre des modèles uniques. Le second est l'identification des relations entre les modèles perçus et le diagnostic possible. Le succès de ces deux étapes est fortement lié aux compétences du clinicien.

Dans un souci d'aide aux médecins, le concept de système informatique dédié à l'aide au diagnostic émerge depuis une quinzaine d'années. Plusieurs équipes de recherche se sont intéressées au développement d'outils pour l'aide à la détection et au diagnostic de toutes lésions. Ils sont destinés à une lecture informatique qui vise à aider mais non à se substituer au clinicien. Leurs principes obéissent aux quatre étapes suivantes, à savoir la numérisation, l'extraction de caractéristiques propres aux images, la classification et le diagnostic.

Les techniques, mises au point, en sont plus à des procédés « d'aide à la détection », ceux qui constituent, à ce stade des outils déjà très intéressant pour le clinicien, puisqu'ils permettent de cibler une zone dite « suspecte », que l'expert peut diagnostiquer.

Le traitement d'images s'avère être un outil permettant une automatisation des tâches qui va assister l'expert aussi bien dans l'analyse qualitative que quantitative des images.

PROBLEMATIQUE

L'interprétation des images IRM cérébrales de sujets pathologiques se fait actuellement de façon manuelle, aussi bien pour la reconnaissance des structures cérébrales ou des lésions, que pour leur caractérisation. Ce travail fastidieux et long dépend d'un certain nombre de paramètres à même de fausser la lecture, les mesures ou même l'interprétation.

L'objectif de ce travail est :

- D'améliorer la qualité des images utilisées ;
- D'extraire la lésion avec le plus de précision possible ; de la caractériser géométriquement pour pouvoir la localiser dans le cerveau, et en déduire des paramètres tels le volume de la lésion.
- D'essayer de générer de façon automatique ou semi-automatique, un certain nombre d'informations objectives, dont le but est de faire une aide au diagnostic ou au geste opératoire, ou le suivi temporel de pathologies.

Il s'agit donc de concevoir un système de segmentation d'Images de Résonance Magnétique (IRM) cérébrales, qui permet de tirer partie de plusieurs approches complémentaires en réponse à la complexité du problème posé. Nous avons opté pour une

détection de conteur (qu'il fait partie de la segmentation) par Bandelettes des coupes IRM, précédée en amont d'une phase de prétraitements adéquats et adaptés aux acquisitions IRM.

Un grand nombre d'outils ont été utilisés pour cette tâche comme la morphologie mathématique, la diffusion anisotrope, la segmentation d'images.

L'objectif du projet proposé par l'équipe de recherche AC2 (Architecteurs pour la classification et cryptographie) de la division ASM du Centre de Développement des Technologies Avancées (CDTA) est l'implémentation de la méthode des Bandelettes par l'algorithme *Lifting Schème* par deux architectures pour une meilleure optimisation.

Afin d'atteindre notre objectif, nous avons organisé notre mémoire selon les étapes suivantes :

Le premier chapitre donne des généralités sur l'IRM cérébrales, la segmentation et les ondelettes

Le deuxième chapitre présente la détection de contour par la méthode de bandelettes sur l'image médicale.

Le troisième chapitre porte sur la méthode de *Lifting Schème* et son implémentation software sous l'outil MATLAB.

La dernière partie est consacrée à l'implémentation hardware du système embarqué sur un circuit programmable FPGA type ZedBoard et les résultats d'implémentation ainsi que les performances temporelles.

Nous terminons par une conclusion illustrant les perspectives de recherche et développement de ce projet.

1. Chapitre 1 : généralité sur l'IRM cérébrales, la segmentation et les ondelettes

1.1. Introduction :

Le but du traitement d'images médicales est d'extraire à partir des images acquises des informations utiles au diagnostic ou de révéler des détails difficiles à percevoir à l'œil nu. Pour cela, le traitement fait appel à des outils, des algorithmes, qui permettent d'agir sur l'image numérisée ; l'un des processus fondamentaux dans la chaîne de traitement d'image est la segmentation.

Dans ce premier chapitre, nous allons commencer par une définition sur les images à résonance magnétique qui représente la modalité choisie dans notre travail ensuite nous allons faire un état de l'art sur la segmentation des images médicales.

À la fin de ce chapitre, nous allons définir globalement les ondelettes qui représentent la méthode adoptée dans notre travail.

1.2. Imagerie à Résonance Magnétique

L'imagerie à Résonance Magnétique (IRM) permet d'obtenir des informations sur l'intérieur d'un objet fermé, en le soumettant à un champ magnétique qu'on fait varier. Elle est basée sur la propriété de certains noyaux d'atomes (en particulier l'Hydrogène), qui possèdent un moment magnétique angulaire de spin.

Placés dans un champ magnétique constant, ces spins tournent à une fréquence qui leur est propre (la fréquence de Larmor). L'orientation de ces spins est soit parallèle au champ, soit inverse à ce champ. A l'équilibre, il y a plus de spins dans l´état parallèle. Par conséquent, le moment magnétique global a une composante transverse nulle (car tous les spins sont d´emphases), et une composante longitudinale non nulle, orientée comme le champ.

Si maintenant l'on fait varier le champ et qu'on excite les atomes à leur fréquence de résonance, on tend d'une part à mettre en phase tous les spins, ce qui crée une aimantation transversale, et on équilibre le nombre de spins dans l´état parallèle et dans l'état anti parallèle, ce qui tend à annuler l'aimantation longitudinale.

Lorsque l'on arrête l'excitation électromagnétique, les spins reviennent à leur état d'équilibre : la composante transversale revient exponentiellement (dans les liquides) à 0 en un temps caractéristique T2, tandis que la composante longitudinale augmente et retrouve sa valeur initiale, en un temps caractéristique T1.

On fait varier la fréquence de l'excitation électromagnétique, afin de faire réagir les différents atomes présents à l'intérieur de l'objet. L'analyse, par transformée de Fourier, des sinusoïdes amorties en temps caractéristiques T1 et T2 permet de connaitre sa nature chimique ; On peut obtenir ainsi, en translatant un patient, toute une série de coupes en contraste, images en tranche de l'intérieur de son corps, et ceci sans avoir à pratiquer aucune intervention chirurgicale. Selon que l'on utilise une pondération T1 ou T2, les zones mises en valeur sont différentes : en pondération T2, l'eau, les œdèmes sont en hyper signal et apparaissent donc nettement. En analysant le temps de relaxation T1, la substance blanche apparait en blanc.

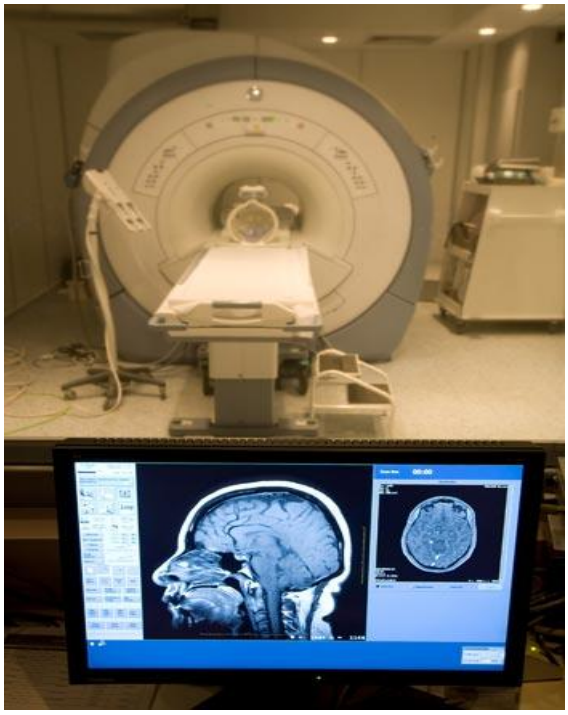

Figure 1.1 Appareil d'IRM de l'antenne tête réceptrice

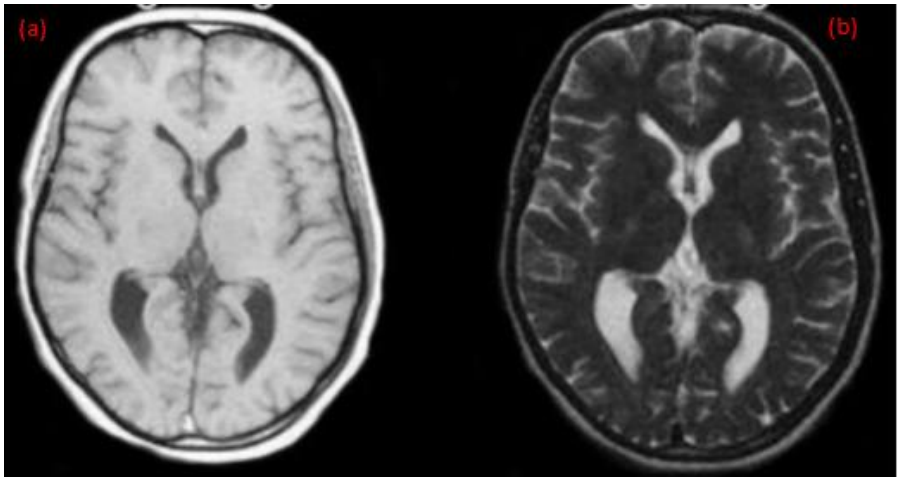

Figure 1.2: (a) Image IRM pondérée en T1. (b) Image IRM pondérée en T2 [LEC 08].

1.2.1. Inconvénients des images IRM (*Artefacts*)

Outre les trois paramètres principaux qui entrent en jeu dans la formation d'une image de résonance magnétique, un certain nombre d'autres facteurs viennent affecter la qualité des images. Les erreurs (artefacts) d'acquisition en IRM sont de natures différentes de celles observées dans d'autres domaines du traitement d'images [SEM 07].

On distingue essentiellement quatre effets: le bruit, le mouvement, les variations de champ et les effets de volume partiel [GER 99]

a. Bruit

Le bruit d'image est la présence d'informations parasites qui s'ajoutent de façon aléatoire aux détails dans les images acquises. [BEL 12]

Dans les images par résonance magnétique, le bruit a des origines multiples, liées en partie au bruit de l'appareillage. L'objectif est d'augmenter le contraste entre les tissus tout en conservant une bonne résolution et un rapport signal/bruit élevé, ces caractéristiques sont cependant contradictoires et il est nécessaire de trouver un bon compromis entre résolution et bruit. Ainsi, on peut doubler la taille des pixels pour multiplier le rapport signal/bruit d'un facteur p donc le choix d'acquisition est donc un facteur déterminant. [MEZ 11]

b. Mouvement

Le mouvement peut provenir de plusieurs sources. Il peut être lié au métabolisme comme la circulation sanguine ou la respiration (déplacement chimique). Il peut également être lié au mouvement du patient pendant l'acquisition. Dans tous les cas, le mouvement diminue la qualité de l'image et pose des problèmes d'interprétation. Les mouvements de la tête, sont responsables d'artefacts dans les IRM cérébrales [MEZ 11].

c. Variations du champ magnétique (inhomogénéité RF)

Les variations de champ ont pour conséquence une variation des intensités d'un même tissu dans une direction quelconque de l'image. Ce phénomène est dû au fait que le champ magnétique n'est pas parfaitement homogène spatialement et temporellement pendant une acquisition. Il existe de plus des non-linéarités de gradient de champ magnétique. [MEZ 11]

Ce biais peut poser des problèmes de classification pour des techniques de segmentation basées sur l'intensité, si on suppose que l'intensité d'une classe est constante sur toute l'image. [BRI 08]

La non-uniformité est prise en compte dans la plupart des méthodes de segmentation, soit en la compensant par prétraitement, soit en la modélisant au cours de la segmentation. [BRI 08]

1.3.La segmentation

La segmentation d'images permet d'apprécier les formes et les contours et de les interpréter de la manière la plus juste possible pour aboutir à une plus grande normativité de l'image sans altérer sa fidélité. [MEZ 11]

La segmentation des structures anatomiques cérébrales est un problème difficile et reste un sujet de recherche de forte actualité dans le domaine médical. La difficulté à segmenter une

image cérébrale vient de la complexité structurelle des images IRM et du contraste souvent insuffisant pour extraire la structure d'intérêt, sans aucune connaissance à priori sur sa forme et sa localisation.

1.3.1. La channe de traitement d'une image médicale :

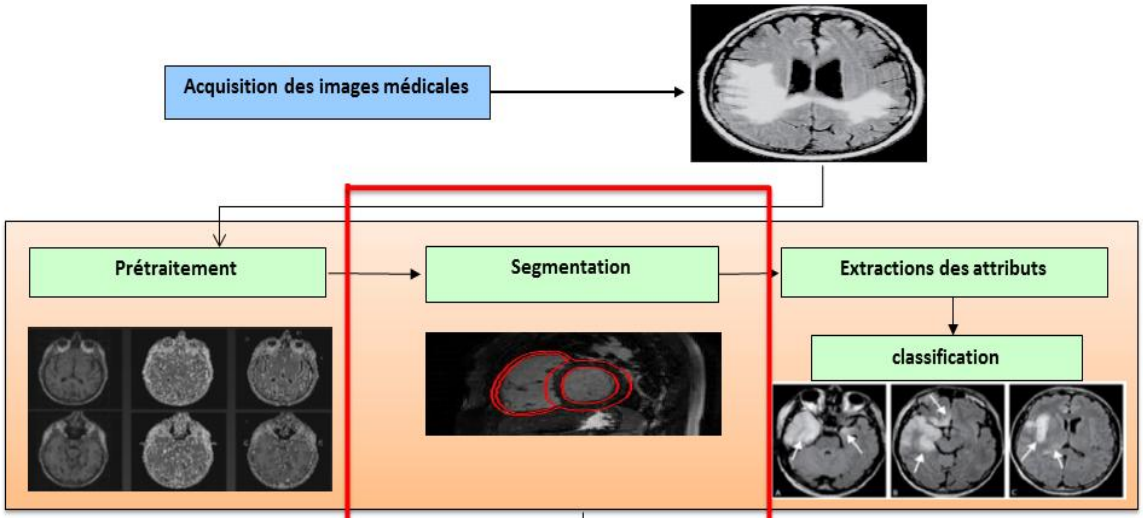

Figure 1.3 : La channe de traitement d'une image médicale

Dans ce mémoire notre objectif est la segmentation est précisément en cherche la meilleure détection de contour.

1.3.2. La segmentation d'images cérébrales

Nous commençons par le but de la segmentation d'images cérébrales et l'apport apporté par la segmentation automatique dans le domaine médical ainsi que les difficultés trouvés dans les images cérébrales.

1.3.2.1. Le But de la Segmentation d'images cérébrales

Segmenter ou délimiter les structures cérébrales est une étape fondamentale pour l'analyse d'images cérébrales. Elle permet de faire une séparation des différents tissus cérébraux (matière grise, matière blanche, liquide céphalorachidien, etc..) ainsi que d'éventuelles pathologies cérébrales [MEZ 11]. Une bonne segmentation permet d'aider le médecin à prendre une décision finale, avant son geste chirurgical.

De nombreuses études des tissus du cerveau dans les images d'IRM ont été effectuées et rapportées dans la littérature, telles que des techniques basées sur les réseaux neuronaux, la logique floue, des méthodes statistiques, coupe de graphe et bien d'autres,

1.3.2.2. La segmentation automatique des images IRM cérébrales

La segmentation manuelle des images IRM est extrêmement couteuse en temps humain en plus l'intervention humaine reste une source potentielle d'erreurs : au cours de longues

séances d'interaction, l'opérateur humain change son comportement, la fatigue le fera commettre des erreurs et le résultat manque de fiabilité et de robustesse.

En comparaison avec la segmentation manuelle qui est par essence une approche de contour, la segmentation semi-automatique, peut être une approche contour ou région. Sur le plan informatique, l'approche région est souvent plus facile que l'approche contour. En effet, il peut être difficile de fermer le contour et donc de quantifier la surface segmentée. Il peut être impossible de déterminer précisément la position du contour car la limite de l'objet n'est jamais nette surtout dans le cas des lésions de sclérose en plaques. Alors que la segmentation entièrement automatique est l'objectif extrême. Elle a de nombreux avantages, en plus d'être indépendante de l'opérateur, elle est rapide et reproductible.

Comme dans le cas de la segmentation semi-automatique, les approches contours où les régions peuvent être envisagées pour la segmentation entièrement automatique.

Donc l'automatisation de la segmentation est nécessaire pour deux raisons :

- Le temps du tracé manuel par un expert sur un tissu spécifique est long,
- L'expertise humaine peut produire des erreurs difficilement contrôlables et reproductibles.

1.3.2.3. Difficultés liées à la segmentation des images cérébrales

La segmentation des IRM cérébrales présente des particularités par rapport aux d'autres domaines d'applications de la segmentation comme la segmentation des cartes routières, ou la segmentation des visages, ceci est dû principalement aux raisons suivantes :

- ✓ L'objet à segmenter (l'anatomie du cerveau).
- ✓ Le processus d'acquisition IRM qui génère des artefacts (défauts).

1.3.2.4. Les différentes approches de segmentation.

La segmentation est un domaine vaste de l'imagerie numérique. Ces dernières années, plusieurs thèses l'ont traitée. Il est tout de même difficile de choisir une méthode parmi les nombreuses qui existent déjà et cela est dû au fait que chaque méthode est fortement liée à un domaine d'application.

Dans ce qui suit nous allons présenter cinq approches qui sont :

- La segmentation utilisant les contours comme critère de décision,
- Celles basées sur les régions,
- Celles basées sur la forme,
- Celles préférant une approche structurelle,

Nous allons donc introduire dans cette section les approches de segmentation les plus connues et nous insistons sur l'approche forme contenant la technique de segmentation par ondelette

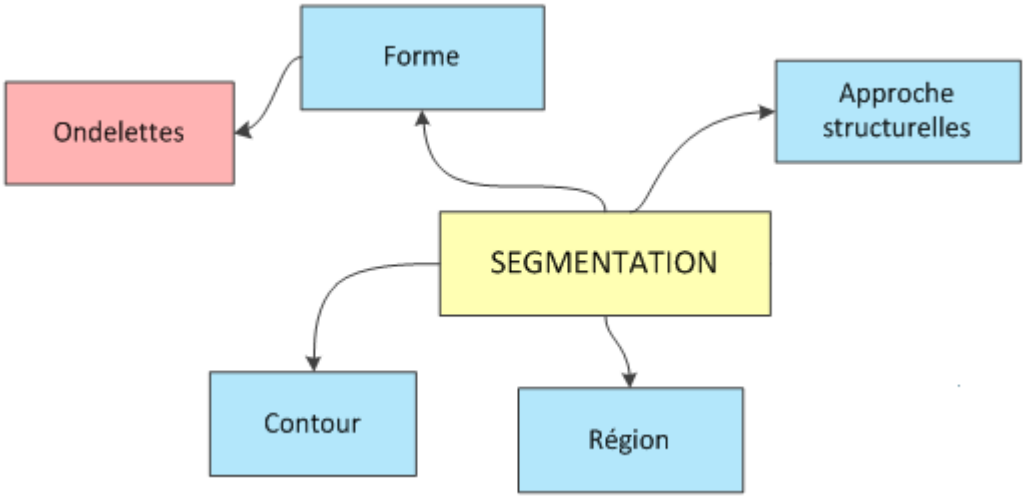

Figure 1.4 : Classification des différentes méthodes de segmentation

a. Segmentations Région

La notion de « région » fait référence à des groupements de point ayant des propriétés communes. Les méthodes de l'approche région aboutissent directement à une partition de l'image, chaque pixel étant affecté à une région unique.

Donc, la segmentation par approche région consiste à rechercher des ensembles de pixels connexes ayant des caractéristiques de luminosité commune [BEL 12].

b. Segmentations contour

La segmentation orientée contours vise à délimiter les objets selon leurs contours. Ces approches ne se basent généralement pas sur les intensités mais sur les variations d'intensité dans l'image, significatives aux frontières entre régions [SCH 08].

c. Approches Structurelles

Dans cette approche, nous allons présenter deux approches les plus célèbres dans la segmentation d'images qui sont : les opérateurs morphologique (Morphométrie = étude quantitative des formes) et la méthode de Ligne de Partage des Eaux (Détection des bassins d'écoulement sur l'image de la norme du gradient) [TRU 98].

d. Segmentations Forme (Modèles déformables)

Les approches basées sur la forme tendent à rechercher des régions qui dérivent d'une forme donnée [MEZ 11].

- ✓ **La méthode des Ondelettes**

C'est une méthode classée dans l'approche **Forme**. L'analyse par ondelettes peut être considérée comme une alternative à la *Transformée de Fourier* et pour résumer : une ondelette est un signal oscillant dont la moyenne est nulle et dont l'énergie tend vers zéro jusqu'à l'infini [LYO 13].

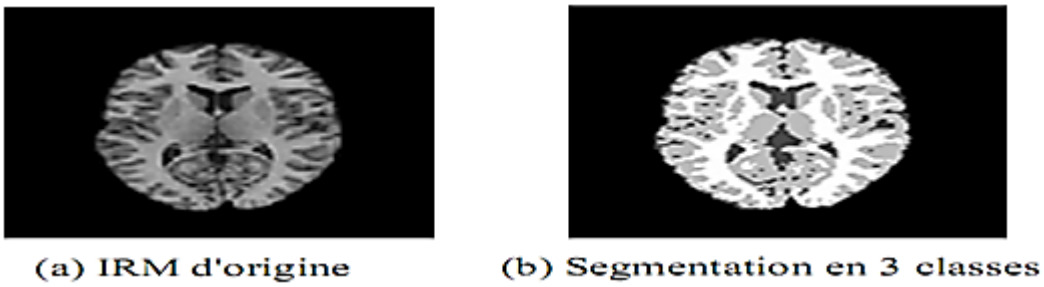

Figure 1.5: Segmentation par ondelettes

1.4.Transformation en ondelette :

1.4.1. Historique

La première transformation en ondelettes - le nom n'est pas encore utilisé - est proposée par ***Haar*** en 1910 ; il serait plus judicieux de parler alors de «pré-ondelette».

La transformée en ondelettes est un outil qui découpe les données, les fonctions sous les opérateurs en composantes fréquentielles suivant une résolution adaptée à l'échelle. Les précurseurs conscients de cette technique ont été des mathématiciens (***Calderon*** 1964), des physiciens (***Aslaken*** et ***Klauderen*** 1968, ***Paul*** en 1985), et surtout des ingénieurs (ou des chercheur de sciences pour l'ingénieur) comme ***Esteban*** et ***Galand*** (1977), ***Smith*** et ***Barnwell*** (1986), ***Vetterli*** (1986). Nous pourions parler dans leur cas de «pré-ondelette». Mais le premier à avoir utilisé la méthode et le premier à avoir proposé le nom d'ondelettes fut ***Jean Morlet*** (1983) [TRU 98].

Le problème traité par ***Morlet*** était celui de l'analyse de données issues des ondes sismiques effectués pour des recherches géologiques ; ces données faites de nombreuses transitoires sont particulièrement adaptées à une technique d'analyse conservant la notion de localisation de l'événement tout en fournissant une information sur son contenu fréquentiel ce qui est tout l'intérêt de ce type de transformation. Les résultats obtenus par ***Morlet*** et formalisés par le physicien ***Alex Grossmann*** ont rapidement éveillé l'attention de nombreux chercheurs et bien tôt des bases mathématiques solides ont été mises en place faisant apparaître la notion de base orthogonale (***Y.Meyer*** 1985), d'analyse multi résolution (***S.Mallat*** 1989) et d'ondelettes à support compact (***I.Daubechies*** 1988). Les ondelettes modernes étaient nées [BER 12].

1.4.2. Intérêt des ondelettes :

La plupart des signaux du monde réel ne sont pas stationnaires, et c'est justement dans l'évolution de leurs caractéristiques (statistiques, fréquentielles, temporelles, spatiales) que réside l'essentiel de l'information qu'ils contiennent. Les signaux vocaux et les images sont à ce titre exemplaire. Or l'analyse de Fourier propose une approche globale du signal, les intégrations sont faites de moins l'infini à plus l'infini, et toute notion de localisation temporelle (ou spatiale pour des images) disparaît dans l'espace de Fourier; il faut donc trouver un compromis, une transformation qui renseigne sur le contenu fréquentiel tout en préservant la localisation afin d'obtenir une représentation temps/fréquence ou espace/échelle du signal [TRU 98].

1.4.3. Théorie de la transformée en ondelettes :

1.4.3.1. Définition :

La transformée en ondelette, d'une fonction f de carré sommable sur Rn est l'ensemble des produits scalaires de cette fonction avec les membres de la famille d'ondelettes continues. La famille d'ondelettes est obtenue par dilatation et translation de l'ondelette mère en variant les deux paramètres d'échelle et de position (a et b) de façon continue sur R (avec la contrainte non nulle).

La transformée en ondelettes s'écrit de manière générale [BER 12]:

$$C_f(a,b) = \int_{\mathbb{R}} f(t)\overline{\Psi_{(a,b)}(t)}dt \qquad \text{eq1.01}$$

Calculer cette fonction$C_f(a,b)$, c'est faire l'analyse de f par l'ondelette ψ. La fonction f est alors décrite par ses coefficients d'ondelettes. Ils mesurent les fluctuations à l'échelle a, de la fonction f.

Cette transformée consiste donc à décomposer un signal en une famille de fonctions localisées en temps et en fréquence.

1.4.3.2. Quelles transformées en ondelettes ?

On peut classer les transformées en ondelettes selon la famille à laquelle appartiennent les fonctions analysantes choisies. Les transformées obtenues sont suivant les cas discrètes ou continues, redondantes ou non [MEZ 11].

1.4.3.3. Transformées en ondelettes Continues :

Les transformées continues sont obtenues en prenant le facteur d'échelle a et le pas de translation b dans l'ensemble des nombres réels, ces transformées sont évidemment très redondantes car l'espace-temps-fréquence est parcouru continûment, ce type de transformation ne peut, dans la pratique, être effectué que de façon approximative et il y a toujours en fait une discrétisation du calcul qui est opérée [TRU 98].

Dans le cas unidimensionnel, la TOC d'un signal f(x) est donc obtenue par l'expression de eq1.02 [TRU 98]:

$$\forall(a,b) \in \mathbb{R}^* \times \mathbb{R} \ , W_f(a,b) = \frac{1}{a}\int_{\mathbb{R}} f(x)\Psi^*\left(\frac{x-b}{a}\right)dx \qquad \text{eq1.02}$$

Avec:

• ψ∗(x) est l'ondelette mère analysante.

• a est le paramètre d'échelle.

• b est le paramètre de position.

• $W_f(a,b)$ est le résultat de la transformée donné par un ensemble des coefficients appelés coefficients d'ondelette de la fonction f(x) dans l'espace-temps-échelle ou espace-échelle.

1.4.3.4. Inversion :

Le signal f peut être totalement reconstruit à partir de sa transformée en ondelette continue en utilisant la formule de la transformée inverse [BER 12]:

$$f(x) = C_\Psi^{-1} \iint_{-\infty}^{+\infty} W_f(a,b)\Psi_{a,b}(x)\frac{da\ db}{a^2} \qquad \text{eq1.03}$$

1.4.3.5. Propriétés de la transformée en ondelette continue :

La transformée en ondelettes continue est caractérisée par les propriétés suivantes :

1. La TOC est une transformation linéaire [DOS 05]:

   • Si f(x)= $f_1(x)$+ $f_2(x)$ alors W_f (a;b)= W_{f1}(a;b)+ W_{f2}(a;b)

   • Si f(x)= k$f_1(x)$ alors W_f (a;b)= kW_{f1}(a;b)

2. LA TOC est covariante par translation:

   • Si $f_0(x)$= f(x − x_0) alors Wf_0(a;b)= W_f (a;b − x_0)

3. La TOC est covariante par dilatation:

   • Si $f_s(x)$= f(sx) alors W_{fs}(a;b)= $\frac{1}{s^{1/2}}$$W_f$ (sa;sb)

1.4.4. Exemples d'ondelettes de base:

Il existe de nombreuses formes d'ondelettes, le choix de l'ondelette optimale dépend de l'application envisagée. Il convient de bien cerner le problème à étudier et d'identifier le type de transformée à utiliser (continue ou discrète). En analyse d'image, il est souvent utile d'avoir une certaine redondance pour avoir plus d'informations. L'utilisation de la transformée en ondelettes continue est alors conseillée. Si on veut un calcul exact, alors les ondelettes à support compact sont indiquées. On voit donc qu'on ne peut parler d'une ondelette idéale, adaptée à tous les cas.

Nous avons choisi de présenter quatre types d'ondelettes qui nous semblent être les plus utilisées dans le traitement du signal: les ondelettes de ***Morlet***, le chapeau mexicain, les ondelettes de ***Haar*** et les ondelettes de ***Daubechies***.

1.4.4.1. L'ondelette de Morlet :

L'ondelette de ***Morlet*** teste l'ondelette complexe la plus fréquemment utilisée. Elle est obtenue en modulant une exponentielle complexe par une enveloppe gaussienne. Elle permet de minimiser le produit des étalements temporels et fréquentiels de l'ondelette, et donc de maximiser la précision de la localisation de l'énergie dans le plan temps fréquence. Elle est définie par [LYO 13]

$$\Psi(x) = e^{i5x} e^{-x^2/2} \qquad \text{eq1.04}$$

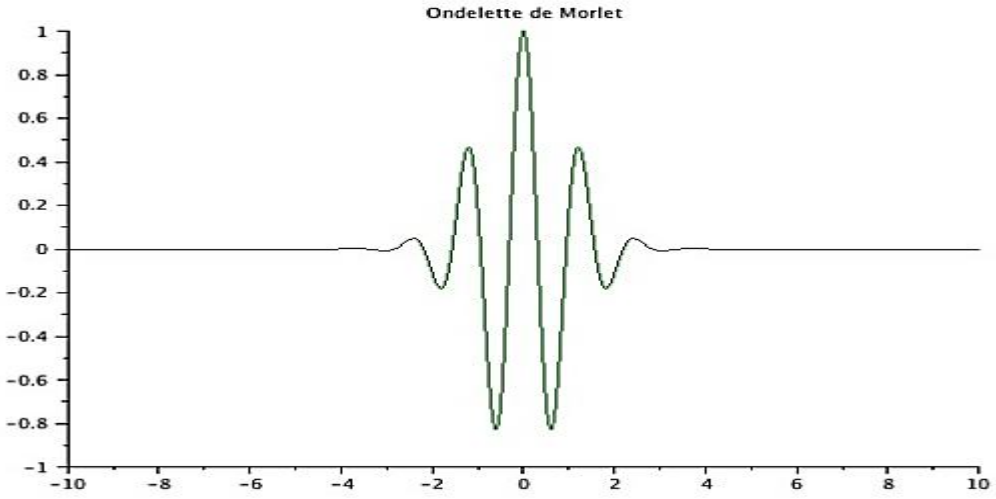

Figure 1.6: Ondelette de Morlet

✓ Propriétés

C'est une ondelette à régularité infinie, symétrique et utilisée dans les transformations continues. Elle est donc bien localisée en espace et en fréquence [LYO 13].

✓ Intérêt

Sa régularité en fait un outil particulièrement adapté pour l'étude de la régularité de fonctions. De plus, son caractère directionnel est très utile pour des signaux tels que les images sismiques [LYO 13].

1.4.4.2. L'ondelette de *Haar* :

Elle est définie par [BER 12]

$$\begin{cases} \Psi(x) = 1 \quad si \;\; 0 < x < \frac{1}{2} \\ \Psi(x) = -1 \;\; si \;\; \frac{1}{2} < x < 1 \\ \Psi(x) = 0 \qquad\quad sinon \end{cases} \qquad \text{eq1.05}$$

La fonction échelle de ***Haar*** :

$$\begin{cases} \emptyset(x) = \emptyset(2x) + \emptyset(2x-1) \\ \Psi(x) = \emptyset(2x) - \emptyset(2x-1) \end{cases} \qquad \text{eq1.06}$$

C'est-à-dire **:**

$$\begin{cases} \emptyset(x) = 1 & si \ \ 0 < x < \frac{1}{2} \\ \emptyset(x) = -1 & si \ \ \frac{1}{2} < x < 1 \\ \emptyset(x) = 0 & sinon \end{cases} \qquad \text{eq1.07}$$

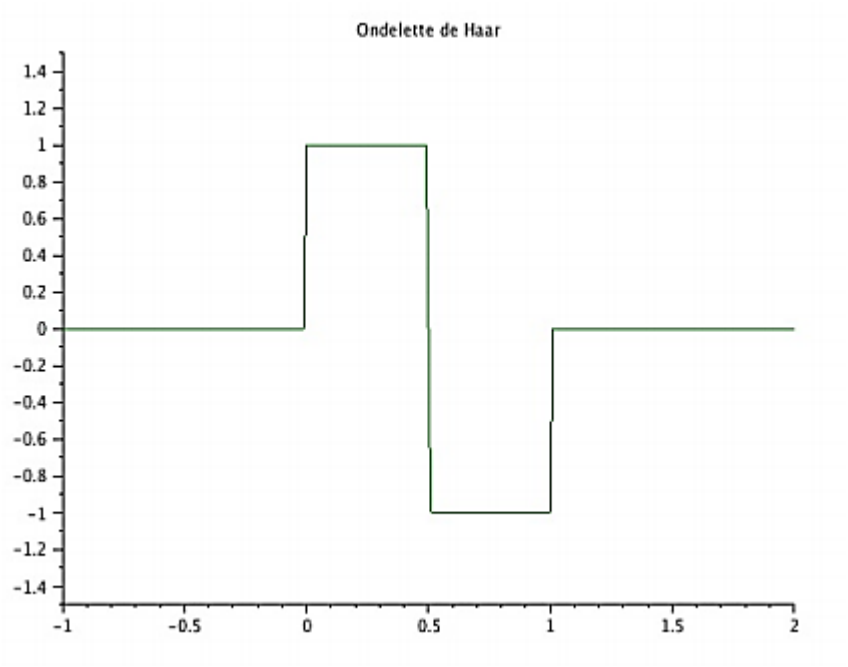

*Figure 1.7: Ondelette de **Haar**.*

- ✓ Propriétés :

C'est une ondelette orthonormale à support compact, symétrique. Elle permet d'obtenir une reconstruction exacte du signal. Elle est utilisée à la fois pour les transformées continue et discrète [LYO 13].

- ✓ Intérêt :

Cette ondelette est très simple et facile à implémenter. De plus elle est à support compact. Le calcul de la transformée de Fourier est donc exact [LYO 13].

1.4.4.3. L'ondelette de *Daubechies* :

L'ondelette de ***Daubechies*** est la famille la plus connue des ondelettes orthonormales. Ses ondelettes sont généralement dénommées par le nombre de coefficients a k non nuls, on parler donc d'ondelettes ***Daubechies4***, ***Daubechies6***,etc.

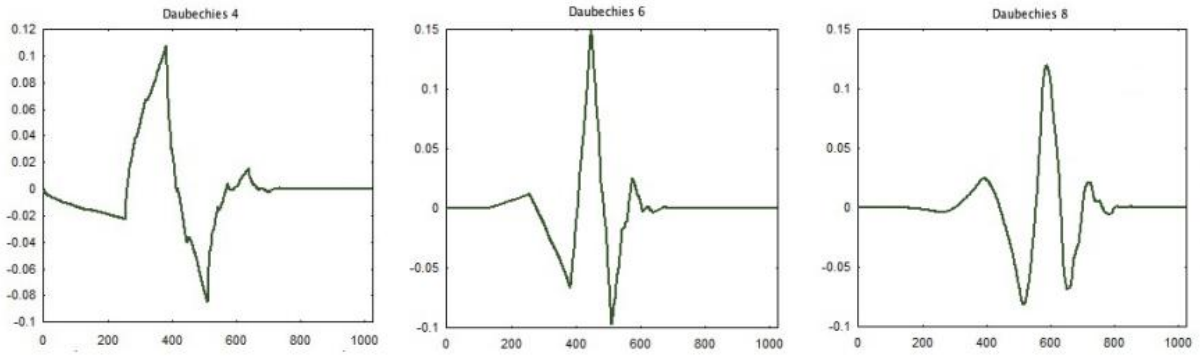

Figure 1.8:Ondelettes de Daubechies.

- ✓ Propriétés :

Quand l'ordre augmente, les supports grandissent ainsi que la régularité des ondelettes [BER 12].

- ✓ Intérêt :

La mathématicienne ***Ingrid Daubechies*** a cherché dans ses travaux à concilier deux contraintes respectives: l'orthogonalité de la base d'ondelettes et la compacité du support de l'ondelette-mère. Ce qui implique que toute ondelette de la base est à support compact et donc que le calcul de la transformée en ondelettes est exacte. De plus, elle a imposé à ses ondelettes une troisième condition: avoir n moments nuls. [TRU 98]

1.5.Transformée en ondelette discrète:

La transformée en ondelettes continues présentée précédemment est obtenue en prenant le facteur d'échelle **a** et le pas de translation **b** dans l'ensemble des nombres réels. Ce type de transformation ne peut être effectué dans la pratique que de façon approximative, et il y a toujours, en fait, une discrétisation du calcul, qui est opérée. ***Morlet*** a formulé des bases construites par une discrétisation dyadique de ces paramètres sur le modèle suivant [NUL 11]:

$$a = 2^{-j}, b = k2^{-j} \quad avec: j = 1,2,\ldots.,J-1 \quad et \quad k = 1,2,\ldots.,2^{-j}-1.$$

Et l'ensemble des fonctions d'ondelettes analysantes seront donc :

$$\Psi_{(a,b)}(t) = \frac{1}{\sqrt{2}}\left(\frac{t-b}{a}\right) \qquad \text{eq1.08}$$

$$\Psi_{(j,k)}(t) = 2^{\frac{j}{2}}.\Psi\left(2^j t - k\right) \qquad \text{eq1.09}$$

Avec $\Psi_{(a,b)}(t) = \Psi_{(j,k)}(t)$ eq1.10

La transformée en ondelettes discrètes (TOD) de la fonction **f**(t) est donc en fonction de **j** et **k**, au lieu de **a** et **b**, respectivement, et entraîne un ensemble de coefficients d'ondelettes (détail) :

$$C_{j,k} = \int_{-\infty}^{+\infty} f(t).\Psi_{(j,k)}(t)dt \qquad \text{eq1.11}$$

Comme les fonctions d'ondelettes, il y a encore un autre ensemble de fonctions appelées fonctions d'échelle φ(**t**) qui donnent par convolution avec **f**(**t**), l'ensemble des coefficients d'approximation :

$$A_{j,k} = \int_{-\infty}^{+\infty} f(t).\emptyset(x)_{j,k}(t).dt \qquad \text{eq1.12}$$

Les fonctions d'ondelettes (détail) et d'échelles (approximation) établissent un algorithme de décomposition multi-résolution. La fonction d'ondelette est orthogonale à la fonction d'échelle à un indice d'échelle particulière **J**. Ainsi, les informations contenues dans les coefficients d'approximation d'un indice d'échelle **J** ne sont pas répétées dans les coefficients d'ondelettes. De plus, les coefficients d'ondelettes à un indice d'échelle donnée sont en fonction de la fonction d'échelle de niveau inférieur. Par conséquent, les fonctions d'approximation à un indice d'échelle donnée peuvent être reconstruites en utilisant la fonction d'approximation et les coefficients de détail de l'indice supérieur [NUL 11].

$$f_j(x) = C_{j+1}(x) + A_{j+1}(x) \qquad \text{eq1.13}$$

Ainsi, les fonctions d'ondelettes et d'échelle ont la capacité d'exprimer une fonction dans une résolution inférieure. La fonction de décomposition de l'équation précédente peut s'écrire sous la forme :

$$f_0(x) = C_1(x) + A_1(x) \qquad \text{eq1.14}$$

$$f_0(x) = C_2(x) + A_2(x) + A_1(x) \qquad \text{eq1.15}$$

$$f_0(x) = C_3(x) + A_3(x) + A_2(x) + A_1(x) \qquad \text{eq1.16}$$

La transformée en ondelettes discrètes (TOD) est devenue un outil très polyvalent de traitement de signal, après l'introduction de la représentation multi-résolutions des signaux basée sur la décomposition en ondelettes en 1987. [NUL 11]

Stéphane Mallat a mis en avant une certaine catégorie de décompositions en ondelettes, qui peuvent être réalisées numériquement en un temps très court, par « une transformée en ondelettes rapide », constituée d'une cascade de filtres passe-bas et passe-haut suivie par des opérations de sous échantillonnages par un facteur de deux.

1.5.1. Extension de transformée en ondelettes pour les images :

Il existe plusieurs façons pour étendre la transformation en ondelettes pour un signal 2D, tel que l'image. Le procédé classique consiste à obtenir la fonction d'échelle 2D φ(t) et la fonction d'ondelette Ψ(t) par le produit tensoriel des fonctions d'ondelettes 1D Ψ(t),Ψ(t) et les fonctions d'échelle φ(t),φ(t). [SOU 12]

La fonction d'ondelette mère 2D correspondante permet d'obtenir les détails dans trois directions, horizontale, verticale et diagonale : [SOU 12]

$$\begin{cases} \Psi^V(x,y) = \varphi(x)\Psi(y) \\ \Psi^H(x,y) = \Psi(x)\varphi(y) \\ \Psi^D(x,y) = \Psi(x)\Psi(y) \end{cases} \qquad \text{eq1.17}$$

1.5.2. Coefficients d'ondelettes en 2D :

Cet ensemble de fonctions fournit une représentation dyadique de l'image, avec trois sous-bandes de coefficients d'ondelettes pour chaque résolution n (n =1..N_I) notées nHL, nLH et nHH, et une unique image d'approximation, notée NlLL. Cette représentation est obtenue par la décomposition de la sous-bande de basse fréquence de chaque niveau n du niveau précèdent (n−1), LL est décomposée ligne par ligne et colonne par colonne ainsi quatre nouvelles sous-bandes sont obtenues : nLL, nLH, nHLet nHH. [QUE 08]

1. nLL regroupe les coefficients de basse fréquence selon l'axe vertical et selon l'axe horizontal.
2. nHL regroupe les coefficients de haute fréquence selon l'axe vertical et de basse fréquence selon l'axe horizontal.
3. nLH regroupe les coefficients de basse fréquence selon l'axe vertical et de haute fréquence et selon l'axe horizontal.
4. nHH regroupe les coefficients de haute fréquence selon l'axe vertical et selon l'axe horizontal.

Cette décomposition est dit pyramidale est la décomposition la plus utilisée.

Il existe une multitude d'algorithmes de décomposions selon les besoin de traitement [QUE 08].

1. de décomposer d'autres sous-bandes du niveau n–1 que (n−1)LL , et ainsi former des sous-niveaux de décomposition

2. de décomposer une sous-bande uniquement ligne par ligne ou colonne par colonne, et ainsi former uniquement 2 sous-bandes de niveau n pour une sous-bande donnée du niveau n−1.

3. de faire un choix différent pour les points 1.et 2. à chaque résolution n. [MRS 12]

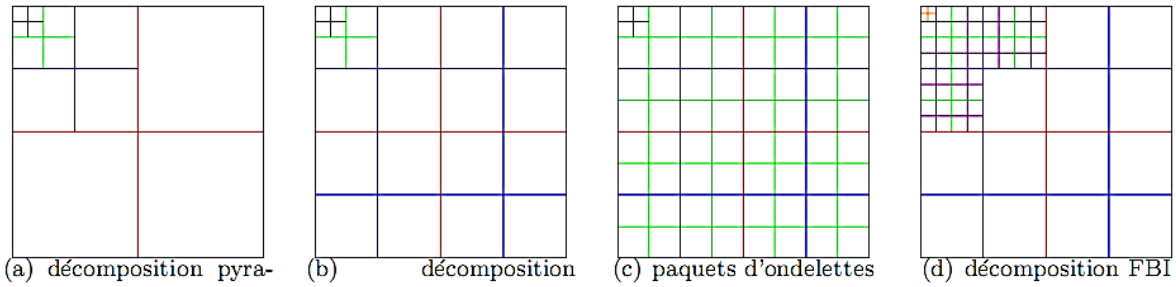

Figure 1.9: Exemples de méthodes de décomposition en ondelette.

1.6.Les Ondelettes géométriques :

Les recherches dans le domaine d'ondelette dirigées vers l'amélioration des possibilités de représentations multi échelles des signaux multidimensionnels. De nombreux travaux donnent plusieurs cadres théoriques de représentations multi échelles plus vastes donnant naissance à de nouvelles transformées plus intéressantes et plus adaptées à l'extraction des structures géométriques lisses et continues telles que les contours. C'est la nouvelle famille

des transformées géométriques qu'ils ont des décompositions multi échelles qui opèrent selon une multitude d'orientations fréquentielles issues d'une décomposition non standard de l'espace des fréquences des images et qui offrent un bon compromis entre la représentation éparse de traits caractéristiques et la qualité des images reconstruites.

Dans la classe des transformées géométriques, deux grandes approches sont proposées: l'approche non-adaptative et l'approche adaptative.

1.6.1. Les transformées géométriques non adaptatives (à base fixe)

Ces transformées utilisent des bases de fonctions fixes mais elles vont chercher à corriger l'aspect isotrope des transformées en ondelettes classiques. On trouve dans cette famille la transformée en Contourlets introduite par ***Minh Do*** et ***Martin Vetterli***. Elle est construite en combinant successivement deux étages de décomposition distincts: une décomposition multi-échelles suivie d'une décomposition directionnelle. On trouve également les travaux de ***Emmanuel Candes*** et ***Dave Donoho*** sur la transformée en *Ridgelets* [CAN 98] [DON 00] dans un premier temps et sur la transformée en Curvelets [CAN 00] dans un second temps, etc.

1.6.2. Les transformées géométriques adaptatives :

Ces transformées définissent une base dont les fonctions sont choisies au mieux pour s'adapter à une image donnée. Pour cela, il est nécessaire d'une étape préalable d'estimation de la géométrie de l'image avant de passer à la décomposition. Dans cette famille il y a par exemple la transformée en Bandelettes (*le détail se trouve dans le chapitre suivant*) dans laquelle les bases sont obtenues par déformation d'une base d'ondelettes selon le flux géométrique local.

1.7.Conclusion :

Dans ce premier chapitre, nous avons présenté brièvement l'imagerie médicale ainsi que la manière dont elle fournit les informations nécessaires sur la forme et le fonctionnement des organes du corps humain. Ensuite, nous avons présenté l'état de l'art sur la segmentation des images médicales et à la fin, nous nous sommes orientés vers les ondelettes qui sont parmi les méthodes de segmentation par formes.

En conclusion, nous pouvons dire que la transformée en ondelettes géométriques est plus adaptée à l'extraction des structures géométriques.

Dans le prochain chapitre, nous effectuons une rétrospective sur les bandelettes qui sont parmi les méthodes basées sur le principe des ondelettes géométriques.

2. Chapitre 2 : la détection de contour par la méthode de bandelettes

2.1. Introduction :

La recherche d'une bonne représentation géométrique des images est un problème central du traitement d'image. On cherche ici un base permettant une représentation creuse des images, c'est-à-dire permettant de bien les approcher avec une détection des contours. Cette propriété est en effet à la base de nombreux algorithmes de représentation d'image parmi eux l'algorithme de bandelettes développé par ***Erwan Le Pennec***, ***Charles Dossal***, ***Gabriel Peyre***, ***Stephane Mallat***.

Nous commençons par l'intérêt de bandelette ensuite nous expliquons la manière d'exploiter les bandelettes pour la détection de contour puis nous présentons l'algorithme d'implémentation de bandelette pour la détection de contour et nous terminons par les résultats..

2.2. Démonstration de détection de contour à partir de bandelettes

2.2.1. Contours, courbes et flots :

2.2.1.1. Contours :

La géométrie de l'image est représentée par ses contours, c'est-à-dire par des courbes. Chaque courbe C peut être paramétrée par son paramétrisation normale, cette paramétrisation se fait par l'abscisse curviligne ***s*** [PEN 02][MAL 05]:

$$C= \begin{cases} \{c(s) = (c_1(s), c_2(s)), s \in \mathbb{R}\} \\ c_1 : \mathbb{R} \to [0,1] \\ c_2 : \mathbb{R} \to [0,1] \end{cases} \quad \text{eq2.08}$$

2.2.1.2. Flot :

La courbe donne une information de direction: sa tangente indique la direction qui ne la traverse pas et qui est donc une direction de régularité maximale. On parle de flot τ le long du contour pour désigner le vecteur unitaire de cette direction de régularité maximale.

$$\tau(c(s)) = c'(s) \quad \text{eq2.09}$$

Telle que $c'(s) = \begin{pmatrix} c'_1(s) \\ c'_2(s) \end{pmatrix}$ eq2.10

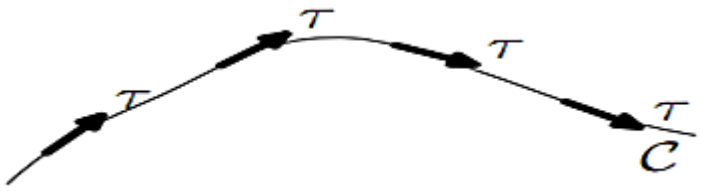

Figure 2.1: Un contour est son flot géométrique associé.

- Lorsque la fonction f est régulière, il s'agit de la direction orthogonale au gradient.

$$\tau(x) = \frac{\nabla f(x)}{|\nabla f(x)|} \qquad \text{eq2.11}$$

- Cette estimation n'est plus possible lors que f n'est plus dérivable, On utilise une technique classique de lissage, La fonction est convolée avec un noyau de lissage d'échelle variable.

$$\tau(x) = \frac{\nabla(g*f)(x)}{|\nabla(g*f)(x)|} \qquad \text{eq2.12}$$

- En effet si le noyau de lissage g est régulier alors la fonction (g*f) est régulière et on peut calculer son gradient:

La nature de l'image (éclairage, contours, texture, etc.)

- Aux opérations en aval de la segmentation (compression, reconnaissance des formes, mesures, etc.)
- Aux primitives à extraire (droites, régions, textures, etc.)
- Aux contraintes d'exploitation (temps réel, espace mémoire, etc.)

2.2.2. Les Bandelettes

- Les bandelettes sont obtenues à partir d'une déformation locale w de l'espace permettant d'aligner la direction de régularité avec une direction fixe [PEN 02][MAL 05]:

- On souhaite trouver une transformation w du plan telle que l'application du gradient ∇w sur la direction τ (x)au point x donne une direction constante τ0:

$$\tau_0 = (\nabla w)(\tau(x)) \qquad \text{eq2.13}$$

- **Déformation de l'espace**

  Le flot τ0 (eq2.13) correspond à la direction tangente à la courbe dans le cas particulier des contours nets. Lorsque ce flot est constant le long des directions verticales, on peut lui associer des courbes intégrales plutôt horizontales et réciproquement à une courbe plutôt horizontale. Ce flot constant τ_0 est associé le long des directions verticales donc aux courbes plutôt horizontales paramétrées par $x_2 = c(x_1)$, la déformation w définie par [PEN 02] [MAL1 12]:

$$C = \{(x_1, c(x_1))\} \qquad \text{eq2.14}$$

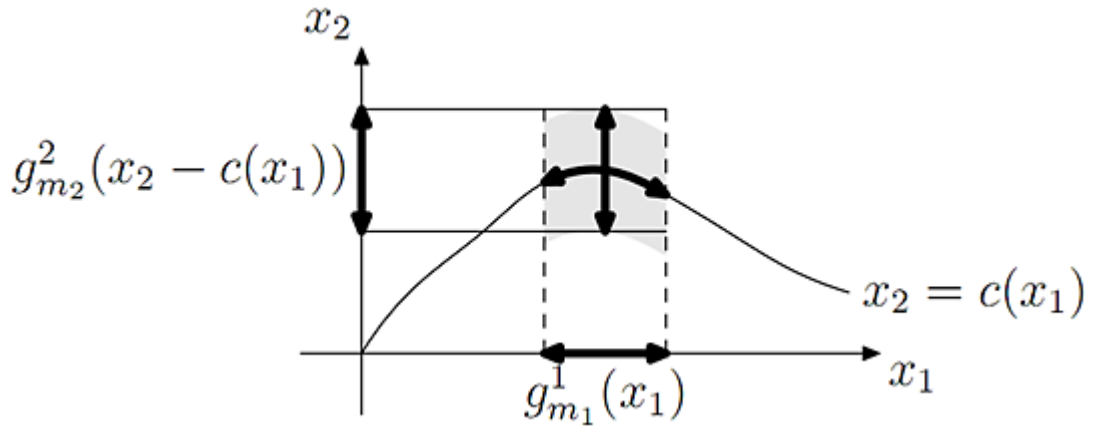

Figure 2.2: Une bandelette au voisinage d'un contour

$$x_3 = c(x_1) \qquad \text{eq2.14}$$

Alors la transformation w est :

$$w: \begin{cases} \mathbb{R}^2 \rightarrow \mathbb{R}^2 \\ (x_1, x_2) \rightarrow (x_1, x_2 - c(x_1)) \end{cases} \qquad \text{eq2.15}$$

Cette seconde transformation impose une limitation sur la tangente de la courbe: celle-ci ne doit jamais être verticale afin de contourner cette limitation, on doit segmenter l'image de sorte que dans chaque morceau la courbe est plutôt horizontale et n'admet pas de tangente verticale et on utilise la transformation W, soit la courbe est plutôt verticale et n'admet pas de tangente horizontale et on utilise la transformation W de l'équation (eq2.15) en inversant le rôle des deux directions.

Si le flot original ne présente pas cette caractéristique, on l'approche pour s'y ramener soit à la courbe horizontale ou verticale.

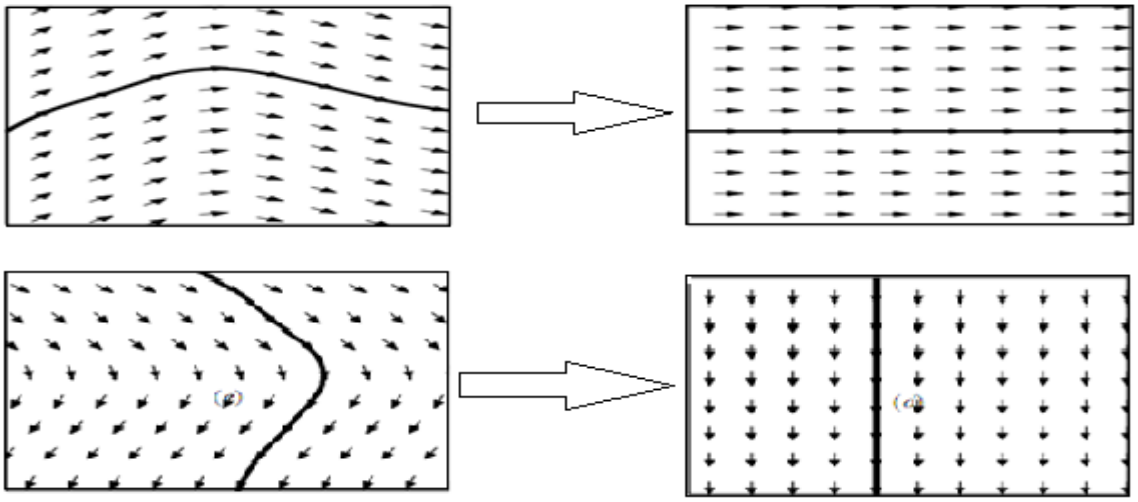

Figure 2.3: Déformation horizontale et déformation verticale.

La transformation w est inversible, conserve la norme et permet d'aligner le flot associé à la courbe avec l'horizontale. Cette déformation induit une W sur les images définies par :

$$Wf(w(x)) = f(x) \qquad eq2.16$$

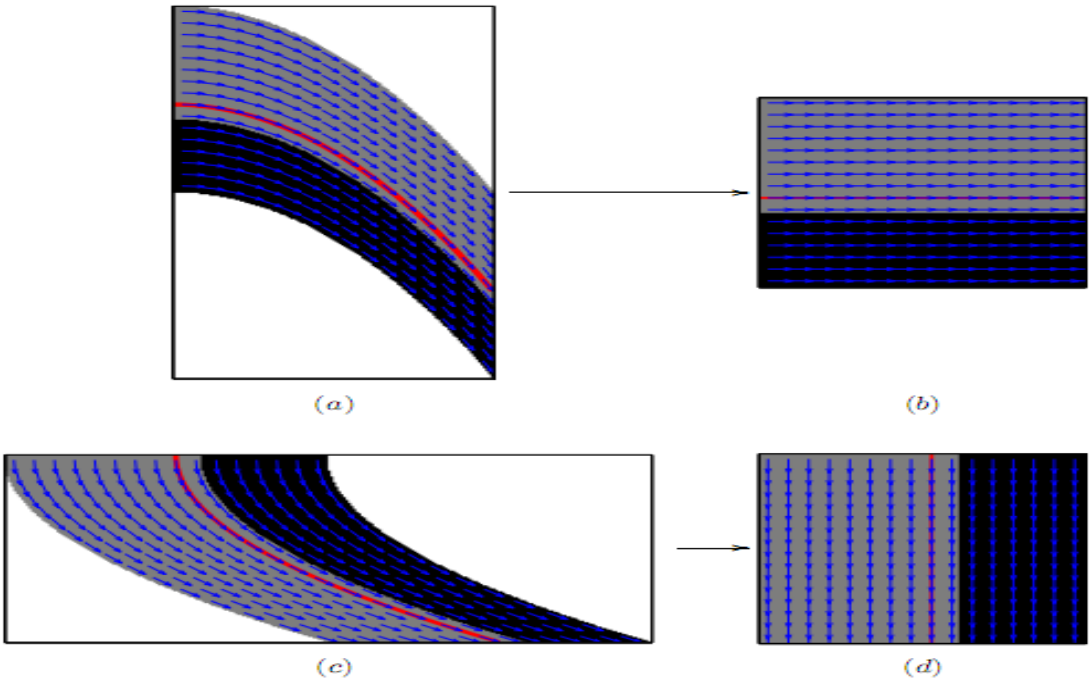

Figure2.4:Exemples de déformation d'un contour (a)Contour vertical (c) Contour orthogonal. (b) Déformation verticale (d) Déformation orthogonale.

Selon *Erwan Le Pennec,* la base de bandelettes de l'espace entier est obtenue comme l'image par W* d'un produit tensoriel de bases orthonormées:

2.2.3. Démonstration de bases de Bandelettes

- On souhaite maintenant introduire des bases utilisant la régularité géométrique. Pour cela, on se ramené à la fonction W_f qui est régulière selon la direction horizontale et potentiellement irrégulière selon la direction verticale. On décompose alors celle-ci dans une base séparable B [PEN 02] [MAL 05]:

$$B = \{g^1_{m_1}(x_1) g^2_{m_2}(x_2)\}_{m_1 \in \mathbb{Z}, m_2 \in \mathbb{Z}} \qquad eq2.17$$

- Ou les familles $\{g^1_{m1}\}_{m1 \in Z}$ et $\{g^2_{m2}\}_{m2 \in Z}$ sont deux bases monodimensionnelles. Ceci permet de traiter différemment les deux directions :

$$\langle Wf, g^1_{m_1} \otimes g^2_{m_2} \rangle \qquad eq2.18$$

- On se ramené à des produits scalaires avec f à l'aide de la transposée de l'opérateur W :

$$\langle Wf, g^1_{m_1} \otimes g^2_{m_2} \rangle = \langle f, W^*(g^1_{m_1} \otimes g^2_{m_2}) \rangle \qquad eq\ 2.19$$

Avec :

$$W^*(g^1_{m_1} \otimes g^2_{m_2})(x_1, x_2) = g^1_{m_1}(x_1) g^2_{m_2}(x_2 - c(x_1)) \qquad eq2.20$$

- Soit en insérant l'expression de W_f :

On a:

$$w: \begin{cases} \mathbb{R}^2 \rightarrow \mathbb{R}^2 \\ (x_1, x_2) \rightarrow (x_1, x_2 - c(x_1)) \end{cases} \quad eq2.21$$

Donc:

$$\langle Wf, g^1_{m_1} \otimes g^2_{m_2} \rangle = \int_x f(x_1, x_2 + c(x_1)) g^1_{m_1}(x_1) g^2_{m_2}(x_2) dx \quad eq2.22$$

- Puis par un changement de variable en x_2

$$\langle f, W^*(g^1_{m_1} \otimes g^2_{m_2}) \rangle = \int_x f(x_1, x_2) g^1_{m_1}(x_1) g^2_{m_2}\big(x_2 - c(x_1)\big) dx \quad eq2.23$$

$$\langle f, W^*(g^1_{m_1} \otimes g^2_{m_2}) \rangle = \langle Wf, g^1_{m_1} \otimes g^2_{m_2} \rangle = \langle f, g^1_{m_1}(x_1) g^2_{m_2}(x_2 - c(x_1)) \rangle \quad eq2.24$$

Si on remplace les bases orthonormées $g^1_{m_1} et\ g^2_{m_2}$ dans l'espace isotrope par des bases d'ondelette ***HAAR*** par exemple on obtient la relation suivante :

$$B_c = \left\{ W^*(\Psi_{j_1,k_1} \otimes \Psi_{j_2,k_2} \right\}_{(j_1,k_1)\in\mathbb{Z}^2,(j_2,k_2)\in\mathbb{Z}^2} \quad eq2.25$$

$$B_c = \left\{ \Psi_{j_1,k_1}(x_1) \Psi_{j_2,k_2}(x_2 - c(x_1)) \right\}_{(j_1,k_1)\in\mathbb{Z}^2,(j_2,k_2)\in\mathbb{Z}^2} \quad eq2.26$$

Donc on notera la base d'ondelettes bidimensionnelles du domaine Ω construite à partir de l'ondelette Ψ et de la fonction d'échelle ϕ correspondante [PEN 02]

$$\begin{cases} \phi_{j,m_1}(x_1) \Psi_{j,m_2}(x_2) \\ \Psi_{j,m_1}(x_1) \phi_{j,m_2}(x_2) \\ \Psi_{j,m_1}(x_1) \Psi_{j,m_2}(x_2) \end{cases}_{(j,m_1,m_2)\in I_\Omega} \quad eq2.27$$

Les bandelettes permettent d'exploiter la régularité géométrique des contours pour suivre la régularité des images, donc au lieu de représenter la géométrie des images par des contours, qui sont bien souvent mal définis, celle-ci est définie par des flots géométriques.

Les vecteurs d'un flot géométrique indiquent des directions où l'image a localement des variations régulières. Les bases orthonormées de bandelettes sont construites en divisant l'image en régions où ces flots restent parallèles. La détermination effective de ce flot géométrique est repoussée à la section suivante. Les bandelettes sont construites à partir d'ondelettes bidimensionnelles déformées le long du flot géométrique.

2.2.4. La détermination du flot géométrique

Dans une région Ω le flot géométrique est un champ de vecteur τ (x_1; x_2) qui donne une direction dans laquelle la fonction f est régulière dans un voisinage. Pour construire des bases orthonormées, une condition de parallélisme doit être imposée sur le flot.

Celui-ci est choisi soit parallèle verticalement, $\tau(x_1; x_2) = \tau(x_1)$, soit parallèle horizontalement, $\tau(x_1;x_2) = \tau(x_2)$. Pour permettre plus de flexibilité, cette condition de parallélisme n'est imposée que dans des sous-régions Ω_i de Ω. [MAL1 02]

La région est donc partitionnée en des régions Ω_i munies d'un flot parallèle soit verticalement soit horizontalement Dans le cas où la fonction f est uniformément régulière sur une région Ω_i , le flot géométrique perd son sens et n'est pas définie. En pratique, on se restreindra à des partitions en carrés dyadiques.

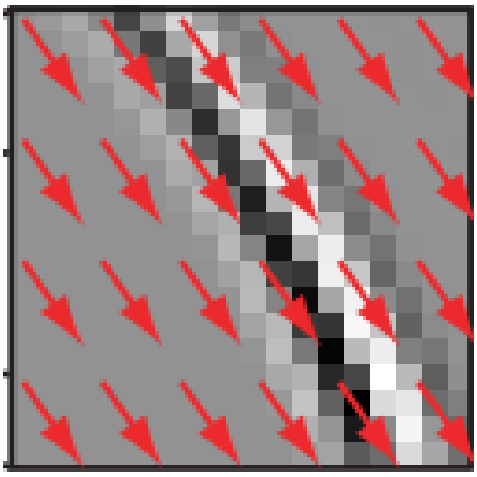

Figure 2.5:Une région Ω avec les flots géométriques correspondants.

On définit alors une base pour chaque région Ω. Si aucun flot n'est défini, on utilise une base d'ondelettes bidimensionnelles de $L^2(\Omega)$.

La construction de cette base est maintenant présentée dans le cas où le flot est parallèle verticalement, le cas horizontal étant similaire.

Le flot s'écrit donc :

$$\tau(x_1; x_2) = \tau(x_1) \qquad eq2.28$$

Et quitte à le normaliser, on peut l'écrire sous la forme:

$$\tau(x_1) = (1 ; c'(x_1)) \qquad eq2.29$$

On pose alors :

$$X_{min} = \inf x_1\{(x_1; x_2) \in \Omega\} \qquad eq2.30$$

Et on définit la ligne de flot comme une courbe intégrale du flot. Celle-ci satisfait l'équation

$$x_2 = c(x_1)+c_0 \qquad eq2.31$$

Où :

$$c(x) = \int_{x_{min}}^{x} c'(u)du \qquad eq2.32$$

c_0 est un paramètre de translation. Par construction, l'image a une variation régulière le long de ces lignes.

En exploitant cette régularité, les ondelettes bidimensionnelles sont déformées pour suivre le flot. L'image déformée :

$$W_f(x_1; x_2) = f(x_1;\ x_2+c(x_1)) \qquad eq2.34$$

est régulière le long des lignes horizontales (x_2 fixé). On utilise alors pour la région déformée [30] [18]. .

$$\Omega' = W\Omega = \{(x_1;\ x_2) : (x_1; x_2+c(x_1)) \in \Omega\} \qquad eq2.35$$

La base d'ondelettes bidimensionnelles de $L^2(\Omega')$:

$$\begin{cases} \phi_{j,m_1}(x_1)\Psi_{j,m_2}(x_2) \\ \Psi_{j,m_1}(x_1)\phi_{j,m_2}(x_2) \\ \Psi_{j,m_1}(x_1)\Psi_{j,m_2}(x_2) \end{cases}_{(j,m_1,m_2)\in I_{\Omega'}} \qquad eq2.40$$

L'opérateur de déformation W étant orthogonal, l'application de son inverse à ces ondelettes donne une base orthonormée de L^2 que l'on appelle base d'ondelettes déformées :

$$\begin{cases} \phi_{j,m_1}(x_1)\Psi_{j,m_2}(x_2 - c(x_1)) \\ \Psi_{j,m_1}(x_1)\phi_{j,m_2}(x_2 - c(x_1)) \\ \Psi_{j,m_1}(x_1)\Psi_{j,m_2}(x_2 - c(x_1)) \end{cases}_{(j,m_1,m_2)\in I_{\Omega}} \qquad eq2.41$$

Afin d'exploiter la régularité de la fonction f selon le flot, on remplace les ondelettes déformées de la forme suivante :

$$\{\phi_{j,m_1}(x_1)\Psi_{j,m_2}(x_2 - c(x_1))\}_{m_1,m_2} \qquad eq2.42$$

Par la famille de fonctions engendrant le même espace.

$$\{\Psi_{l,m_1}(x_1)\Psi_{j,m_2}(x_2 - c(x_1))\}_{m_1,m_2,l>j} \qquad eq2.43$$

Cette opération est appelée « *bandelettisation* » et on vérifie qu'elle s'implémente par une simple transformée en ondelettes monodimensionnelles discrètes.

Les fonctions $(\Psi_{l,m_1}(x_1)\Psi_{j,m_2}(x_2 - c(x_1))$ sont appelées bandelettes du fait de leurs supports allongés selon les lignes de flots. La base orthonormée de bandelettes du domaine est donc définie par une partition donnée et ces flots géométriques correspondants. On définit ainsi une base orthonormée de bandelettes ou d'ondelettes (s'il n'y a pas de flots) [MAL1 02].

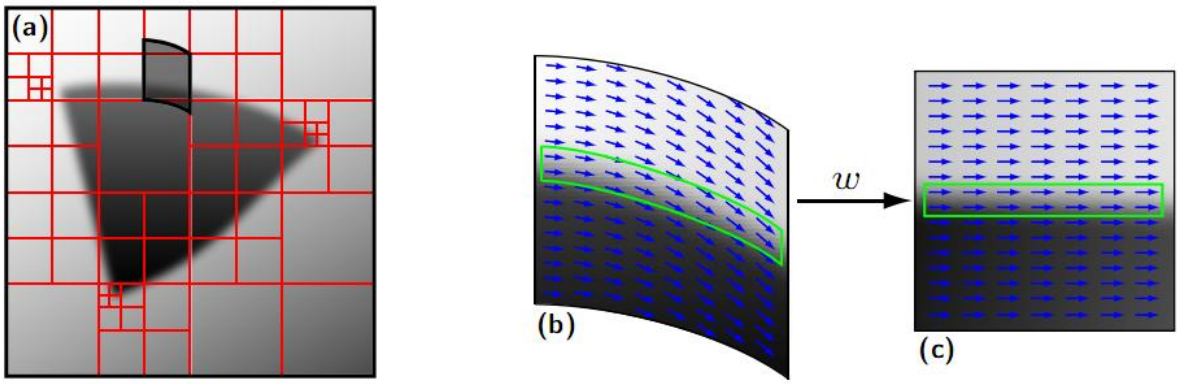

Figure 2.6:(a) Les partitions de bandelette,(b)Les flots géométriques, (c)Déformation de la partition.

La transformation en bandelettes et son inverse s'implémentent par un algorithme rapide de type banc de filtres de complexité O(N2) où N2 est le nombre de pixels de l'image. Pour éviter les effets de blocs, on utilise un schéma de lifting adapté permettant de traverser les bords

2.2.5. Transformée en bandelettes première génération.

Les bandelettes de première génération inventées par *Erwan Le Pennec* et *Stephane Mallat* en 2001 s'appuient sur une description des contours ((a) figure 2.7) de l'image pour en identifier, ou plutôt localiser, les singularités.

Ces contours sont, en premier lieu, triés selon deux catégories : majoritairement verticaux ou majoritairement horizontaux. Contrairement aux transformées présentées jusque-là, plutôt que d'adapter les fonctions d'analyse aux singularités de l'image, la transformée adapte des zones de l'image à la transformée en ondelettes séparables 2D (à quelques adaptations près pour permettre la prise en compte de la forme non rectangulaire des zones). Afin de construire ces zones à partir des contours détectés dans l'image, l'algorithme de Grassfire proposé par *Harry Blum* (Lors du traitement d'image, Grassfire transformé est le calcul de la distance d'un pixel à la frontière d'une région. Il peut être décrit comme aux frontières d'une région d'image pour produire des descripteurs tels que le squelette de la région ou de l'axe médian. *Harry Blum* a introduit le concept en 1967) [MAL 89]. ((b) figure2.7) est utilisée. Ensuite, toujours à partir de l'information des contours, des déformations sont associées à chaque zone pour permettre son adaptation aux familles d'ondelettes séparables 2D ((c) figure 2.7).

Après la prise en compte des pixels des zones déformées (correspondant aux coefficients de bandelettes), il reste un ensemble de pixels à coder. Ces pixels sont pris en charge avec l'estimation de l'image initiale uniquement à partir des coefficients de bandelettes. La reconstruction parfaite est assurée par le codage du résidu entre image initiale et image estimée [PEN 02].

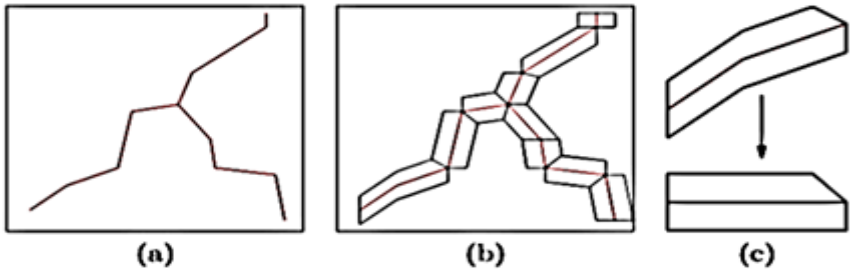

Figure2.7 : Obtention de bandelette de première génération : (a)Détection des contours.(b)Construction des zones des pixels « bandelettes »par Grassfire.(c)Déformation de la zone.

Malgré l'utilisation du Grassfire, la transformée souffre d'une légère redondance qui ne l'empêche pas d'obtenir de bons résultats (notamment grâce à la concentration de l'énergie obtenue par les coefficients de bandelettes). Un autre problème est que l'utilisation de cette transformée de première génération, même si elle respecte les contours, laisse trop peu de détails pour que la restauration apporte une réponse efficace, notamment pour les images contenant de la texture. Afin de rester sur une approche de déformation par zones du

signal, et toujours à partir de la prise en compte des contours de l'image, une deuxième génération de transformée en Bandelettes a été proposée [PEN 02].

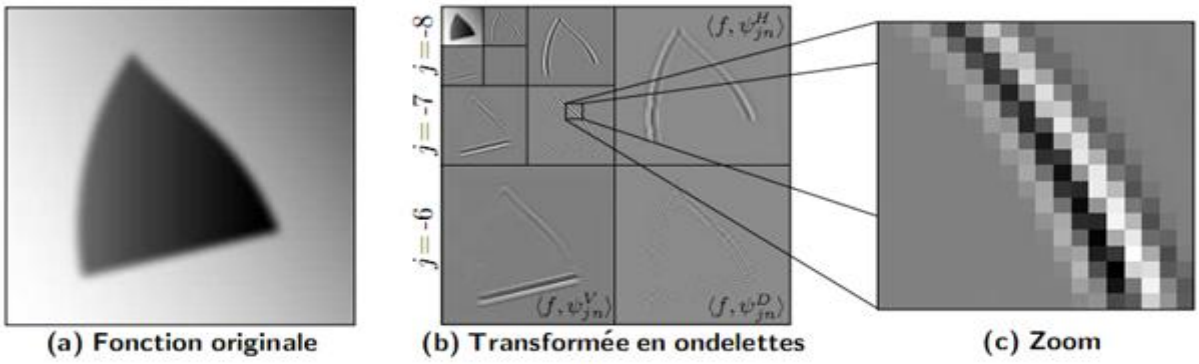

Figure 2.8: Transformée en ondelettes 2D.

2.2.6. Transformée en bandelettes deuxième génération :

Comme nous l'avons déjà remarqué, le schéma d'approximation en bandelettes orthogonales dans la première partie, bien qu'adapté à l'approximation de fonctions géométriques, ne convient pas à l'inversion de l'opérateur de tomographie. En effet, les fonctions bandelettes [PEY 05], ne sont pas localisées en fréquence, ce qui est nécessaire pour bien représenter l'opérateur de tomographie.

En effet l'orthogonalisation de *Gram Schmidt* mélange des fonctions de toutes les échelles. De plus le fait de travailler en espace sur des rectangles crée des ondelettes de bord dont on ne contrôle pas le spectre.

Pour pallier ce problème, un nouveau schéma d'approximation en bandelettes orthogonales de seconde génération est introduit. Cette nouvelle classe de bases orthogonales est construite directement sur les coefficients discrets d'une transformée en ondelettes. Les fonctions de base sont ainsi des combinaisons linéaires d'ondelettes à une échelle fixe. Dans cette section, nous expliquons la construction de ces bandelettes de seconde génération, et énonçons le résultat d'approximation dans une base adaptée de bandelettes qui sera utile pour démontrer l'optimalité de notre estimateur.

2.2.6.1. Transformée en Bandelettes sur un petit carré :

Soit f une fonction avec une régularité géométrique C, 2_j une échelle fixée de transforme en ondelettes, et k une orientation.

Une transformée en bandelettes est implémentée à l'aide d'un ensemble d'opérateurs orthogonaux locaux opérant sur les coefficients en ondelettes de f. Plus précisément, chaque opérateur agit sur un sous ensemble de coefficients $f_s=\{f_j^k[n]\}_{2\wedge j*n\epsilon\ s}$ ou $S=S_1*S_2$ est un carré. [PEY 05]

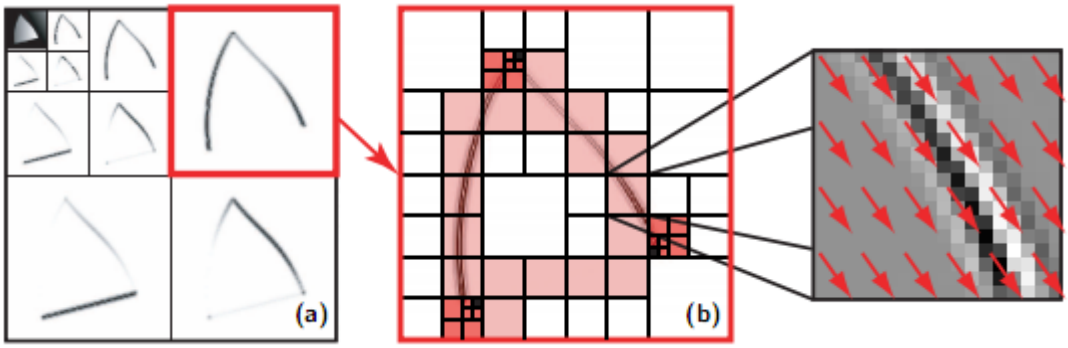

Figure 2.9: Coefficient en ondelettes d'une image.(b)Exemple de segmentation dyadique d'une image géométriquement régulière.(c)Un flot adapté est calculé sur chaque carré.

Une base de bandelettes $B(\lambda)=\{b_v\}_v$ est paramétrée par une géométrie qui spécifie pour chaque échelle et chaque orientation d'une transformée en ondelettes: une segmentation dyadique des coefficients d'ondelettes correspondants, un flot vectoriel indiquant la direction approximative de la géométrie pour chaque carré de la segmentation contenant de l'information géométrique, c'est-à-dire un contour.[MAL 05]

Les bandelettes sont obtenues par un changement de bases orthogonal orienté par cette géométrie sur les ondelettes correspondantes à chacun des carrés de la segmentation. [PEY 05]
Donc les bandelettes permettent de capturer la régularité directionnelle des contours.

L'efficacité de ces bases repose sur l'utilisation de deux algorithmes rapides : le premier permet de calculer la décomposition/reconstruction d'une fonction f dans une base donnée $B(\lambda)$ et le deuxième calcule une base $B(\lambda^*)$ adaptée à f [MAL 05].

2.2.6.2. Algorithme rapide de transformée en bandelettes deuxième génération :

1. L'utilisateur fournit une image discrétisée f de taillée N x N pixels et un seuil T. Une image discrétisée de NXN pixels est obtenue en projetant une fonction *f* sur un ensemble de fonctions d'échelles à une résolution $2^j=N^{-1}$

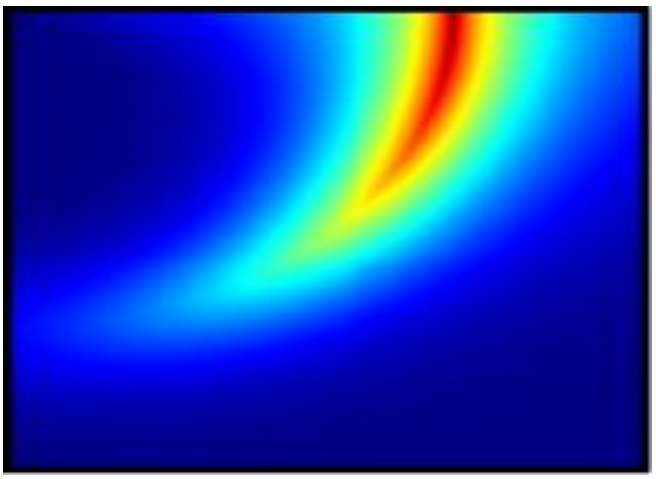

Figure 2.10: Image discrétisée f de taillée N x N pixels.

2. Transformer en ondelette 2D :

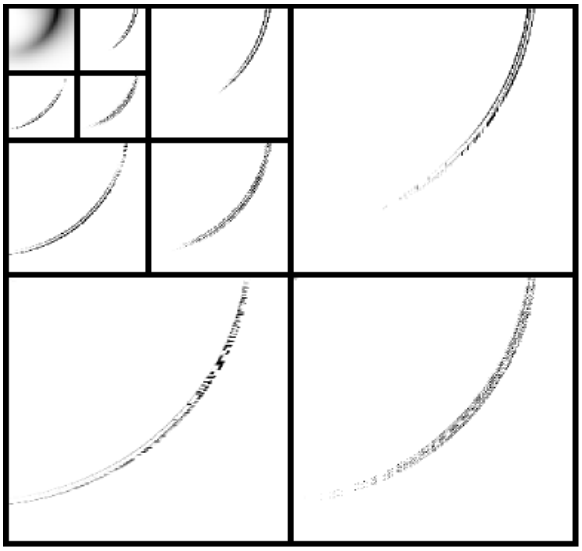

Figure 2.11 : Image Transformer en ondelette 2D

3. Sélection de chaque carré dyadique. Un carré dyadique est par définition un carré obtenu en subdivisant de façon récursive le carré [0; 1] en quatre carrés de même Taille. Pour chaque carré S, on regroupe les coefficients en ondelettes dans un vecteur,

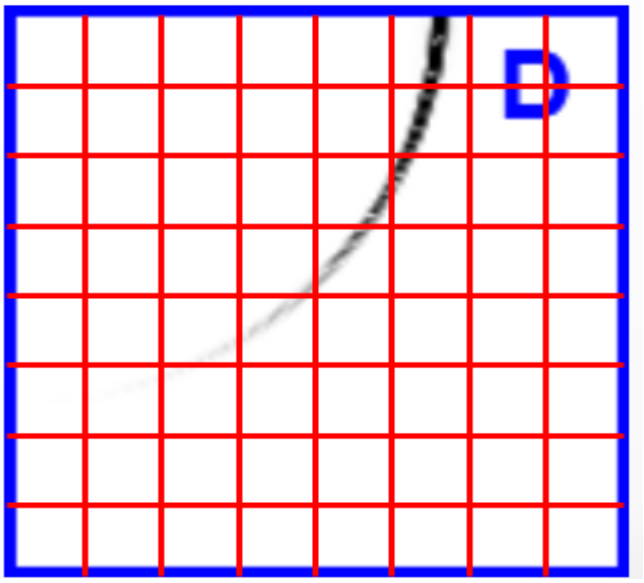

Figure 2.12: Image diviser en carré dyadique.

4. Quantification de la géométrie. Nous devons maintenant trouver la meilleure géométrie approchée quantifiée à l'intérieur d'un carré S. Nous devons donc tester toutes les bases de bandelettes.[MAL 05] [PEY 05]

$\hat{\gamma}_m^{\Omega} = \sum_{i=0}^{p-1}(m_i T^2)\theta_i$ Ou $|m_i| \leq A_\theta T^2$ et $\Omega\{V, H\}$ $eq2.44$

La meilleure sélection géométrique : pour trouver la meilleure direction dans un petit carré, on a besoin de vérifier un ensemble de directions. Le programme sélectionne un nombre de direction ensuite, on les mets dans thêta.

5. Sélection de la meilleure géométrie. Pour un seuil donné T, nous devons choisir La meilleure géométrie qui minimise le Lagrangien. [MAL 05]

$$\mathcal{L}(f_s, B(s,\hat{\gamma}), T) \stackrel{\text{def}}{=} \sum_{(l,i)\in J_t} \left|A_{\hat{\gamma}}(f_s)[l,i]\right|^2 + (Card(J_T) + M_G)T^2 \qquad eq2.45$$

$$J_T \stackrel{\text{def}}{=} \{(l;i) \setminus \left|A_{\hat{\gamma}}(f_s)[l,i]\right| > T \qquad eq2.46$$

- On calcule le lagrangien pour chaque direction, Ensuite on va choisir la direction où le lagrangien est le plus petit.

$$\hat{\gamma}_s \stackrel{\text{def}}{=} argmin(\mathcal{L}(f_s, B(s,\hat{\gamma}), T) \qquad eq2.47$$

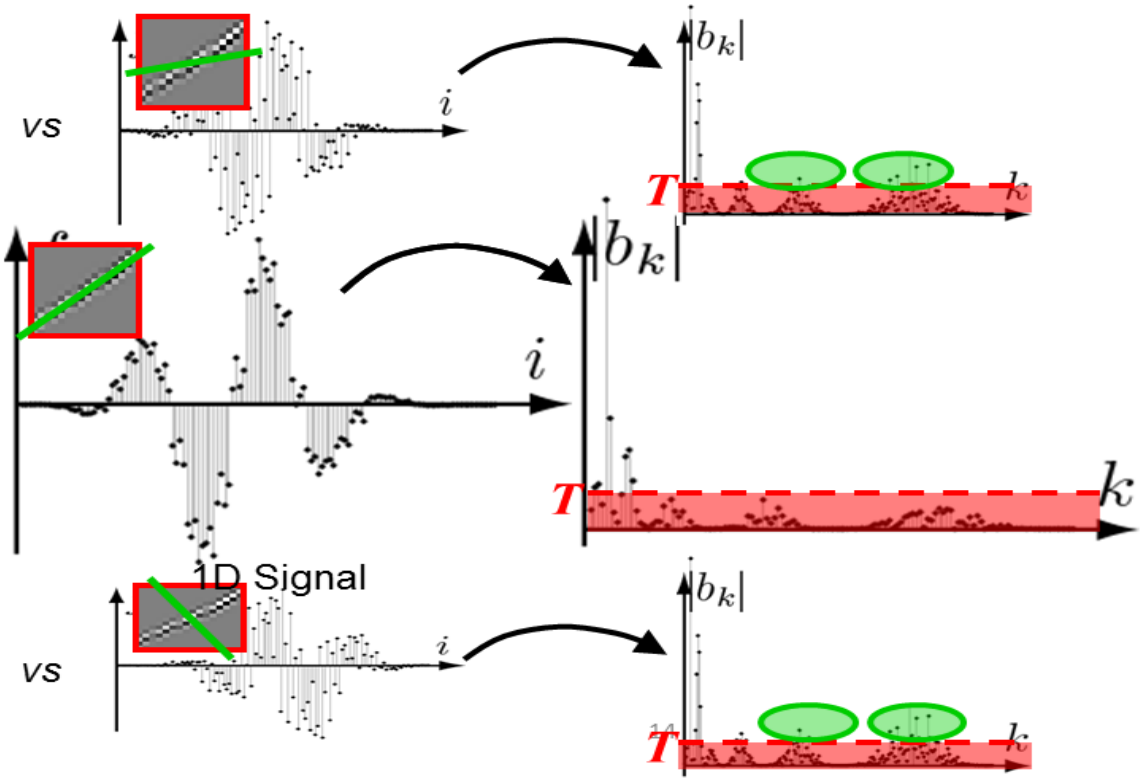

Figure 2.13: Sélection de la meilleure géométrie.

- Pour chaque échelle 2^j Et orientation k, on calcule la structure de Quadtree à l'aide de l'algorithme suivant : [MAL 05]

1. Initialisation du quadtree : chaque carré S de taille b $=2^j$ est une feuille.
2. Enregistrer les géométries optimales S et initialiser L_0, le Lagrangien cumulatif du sous arbre à
   $L_0(S) = L(S)$ *eq2.48*

3. Commencer avec des carrés S de taille b = 2^j Pour chaque carré S, on note (S1; S2; S3; S4) ses quatre sous-carrés et
   L0'(S) = L0(S1)+ L0(S2)+ L0(S3)+ L0(S4)+T^2 *eq2.49*
4. Le Lagrangien du sous arbre (le terme additionnel T2 est dû au coût de un coefficient pour la subdivision). Les sous-carrés doivent être rassemblés si L(S) < L0'(S). Si c'est le cas, déclarer S comme une feuille, enregistrer la géométrie optimale S .Mettre à jour
   L0(S)=min(L(S);L0(S)). Eq2.50
   Tant que b< 1, faire b = 2 * b et répéter l'étape précédente.
5. Transformer en bandelette: Une base de bandelettes $B(k_j)$ de l'espace entier de coefficient lϵ ($[0; 1]^2$) est définie par:
   $$B(\Gamma_j^k) \stackrel{\text{def}}{=} \bigcup_{s \in Q_j^k} B(s, \hat{\gamma}_s) \qquad eq2.51$$

   Un vecteur de bandelettes est ainsi spécifié par :

$$\mu = (j, k, s, \hat{\gamma}_s, l, i) \qquad eq2.52$$

1) 2^j est l'échelle de la transformée en ondelettes 2D et kϵ {V,H,D} est l'orientation
2) S ϵ Qt est un carré dyadique de largeur b =L*2^j où L est une puissance de 2,
3) Thêta est une géométrie approchée,
4) l'échelle et l'index d'un vecteur dans la base de bandelettes orthogonales fait le travail par transformée chaque feuille du quadtree,

- Thêta est une géométrie approchée, l'échelle et l'index d'un vecteur dans la base de bandelettes orthogonales

$$l \in \{-1 log_2(L), \ldots., 0\} \quad et \quad i \in \{1, \ldots., 2^{-l}\}$$

Les vecteurs de bandelettes discrets $\Phi_\mu \in L^2([0,1]^2)$

$$\Phi_\mu = \sum_n \Phi_\mu[n]\, \Psi_{jn}^k \qquad eq2.53$$

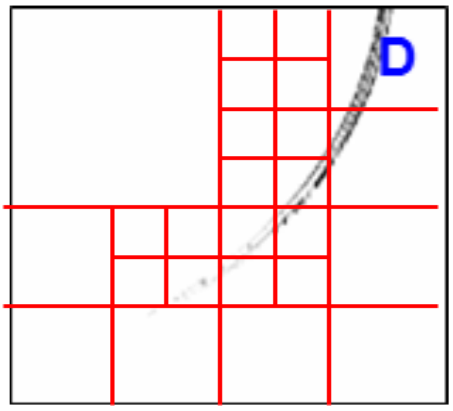

Figure 2.14 : Image transformer en bandelette.

example de bandilitisation

2.3.Implémentation software de la transformée en bandelettes

Pour l'implémentation software de la transformée en bandelettes, nous avons exécuté le Tolboox_bandelette de *Gabriel Peyre* :

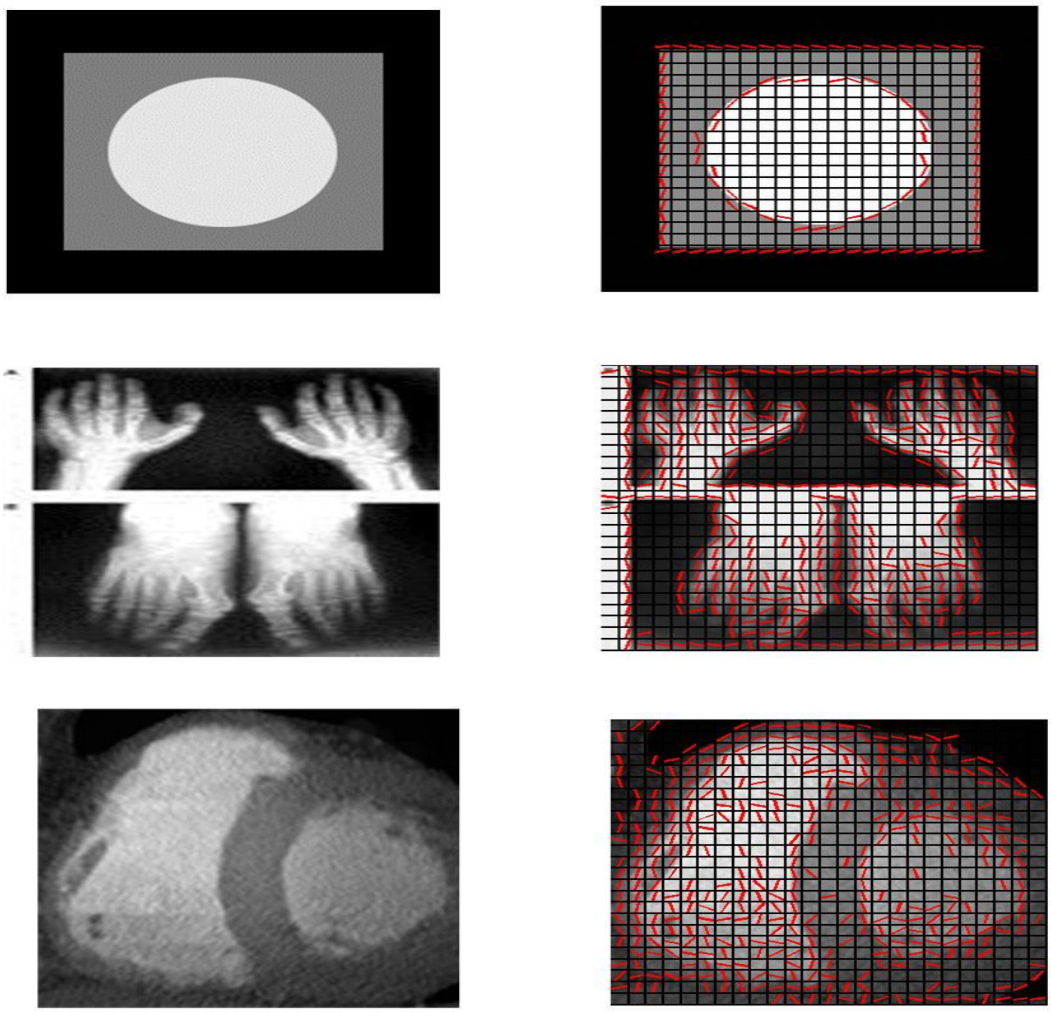

Figure 2.15: Résultats de simulation sur MATLAB.

2.4.Conclusion :

Dans ce chapitre, nous avons défini la représentation géométrique des images à partir de la méthode de bandelettes de façon simplifiée et nous avons commencé par la démonstration de l'adaptation de base bi orthogonale d'ondelette à la détection de contour par l'utilisation des flots géométriques fait par ***Erwan Le Pennec*** et ***Stéphane Mallat*** en 2001 ensuite nous avons présenté l'algorithme de bandilitisation élaboré par ***Gabriel Peyre*** et ***Stephane Mallat*** en 2005.

3. Chapitre 3 : L'Approche *Lifting schème*

3.1. Introduction:

L'analyse de fourrier est le principe de base de la majorité des ondelettes ce qui signifie une difficulté consistante de leur implémentation hardware, c'est qu'en 1994 que ***Wim Sweldens*** [LAN 05] développe l'approche de *Lifting Schème*. Son idée consiste à fournir une relation entre toutes les analyses multi résolution qui partage le même filtre passe haut ou passe bas.

Ce chapitre est consacré donc à l'étude et l'implémentation sous MATLAB de cet algorithme, pour cela nous commençons par des définitions et des présentations théoriques du *Lifting Schème*, par la suite nous implémentons sous MATLAB l'ondelette de ***Haar*** avec l'approche de *Lifting Schème* en deux méthodes différentes.

Le premier est une implémentation par un algorithme dédié que pour l'ondelette de ***Haar*** et la seconde consiste à utiliser l'approche de *Lifting Schème* généralisé qui peut s'adapter à plusieurs filtres (***Daubechies***, ***Haar***…). Et enfin, nous présentons les résultats obtenus à partir de ces deux implémentations.

3.2. Définition de l'Approche de *Lifting Schème*:

Le *Lifting Schème* est un procédé de construction d'ondelettes, mieux efficace que l'approche par banc de filtres. Une de ses caractéristiques les plus importantes est qu'elle permet pour tout banc de filtre basé sur le Lifting de satisfaire automatiquement la propriété de reconstruction parfaite. [JAN 15]

Si le *Lifting schème* est appliqué sur un signal de longueur 2×n, nous obtenons deux signaux de longueur n : un signal d'approximation et un signal de détails.

3.3. Algorithme d'analyse de l'approche de *Lifting schème*

L'algorithme de l'analyse du *Lifting schème* est donné dans la figure3.1. Il se décompose en 3 principales étapes:

- **Séparation (Split) :** Cette étape consiste à découper le signal à analyser (S_j) en deux signaux, le premier signal $even_{j-1}$ contenant les échantillons pairs et le second signal odd_{j-1} contient les échantillons impairs. La taille de chacun des deux signaux obtenus est égale à la moitié de la taille d'origine. Cette étape est appelée transformation en ondelettes paresseuses (*Lazy Wavelet Transform*): [JAN 15]

$$(odd\,, even) = Splite(Sj) \qquad eq3.01$$

- **Prédiction (Predict) :** Considérant que les signaux pairs et impairs sont corrélés, nous pouvons estimer l'un d'eux en connaissant l'autre. un opérateur de prédiction noté (P) est donc défini permettant de prédire les échantillons impairs à partir des échantillons pairs [LAN 05] comme suit:

$$d = odd - P(even) \qquad eq3.02$$

- **Mise à jour (Update) :** Le calcul de la prédiction précédente introduit des erreurs dans le signal (car la prédiction n'est pas exacte). Il convient donc de mettre à jour le résultat précédent grâce à un opérateur de mise à jour U [LAN 05].

$$s = even + U(odd) \qquad eq3.03$$

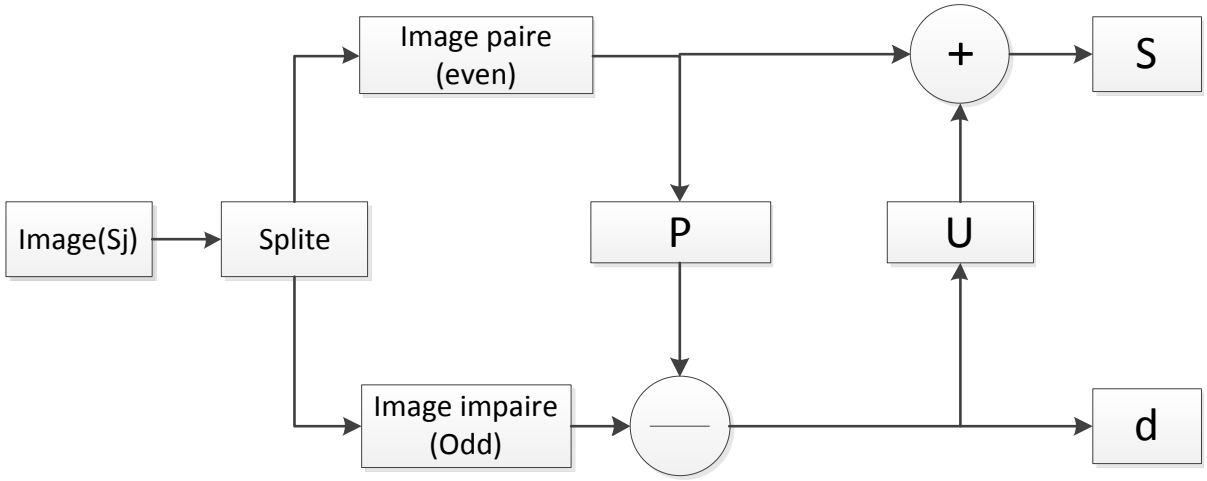

Figure 3.1 : Schéma de L'algorithme du Lifting schème d'analyse.

3.4.Algorithme de synthèse de l'approche de *Lifting schème*

Tout comme les ondelettes classiques, le *Lifting schème* est réversible et son inverse est aussi composé de trois étapes :

- **Mise à jour inverse (Undo update)** : On retrouve le signal pair de départ simplement en inversant le signe de l'opérateur de mise à jour U. [LAN 05]

$$even = S - U(odd) \qquad eq3.04$$

- **Prédiction inverse (Undo predict)** : Connaissant les échantillons pairs et l'opérateur de prédiction P, il suffit d'inverser le signe de l'opération dans l'analyse. [LAN 05]

$$odd = d + P(even) \qquad eq3.05$$

- **Regroupement (Merge) :** regroupe l'ensemble les signaux pairs et impairs en les intercalant. Cette étape est la transformation en ondelettes paresseuses inverse (*Inverse Lazy Wavelet Transform*). [LAN 05]

$$S(j) = Merge(odd\ , even) \qquad eq3.06$$

Le schéma de synthèse *Lifting schème* est donné sur la figure suivante :

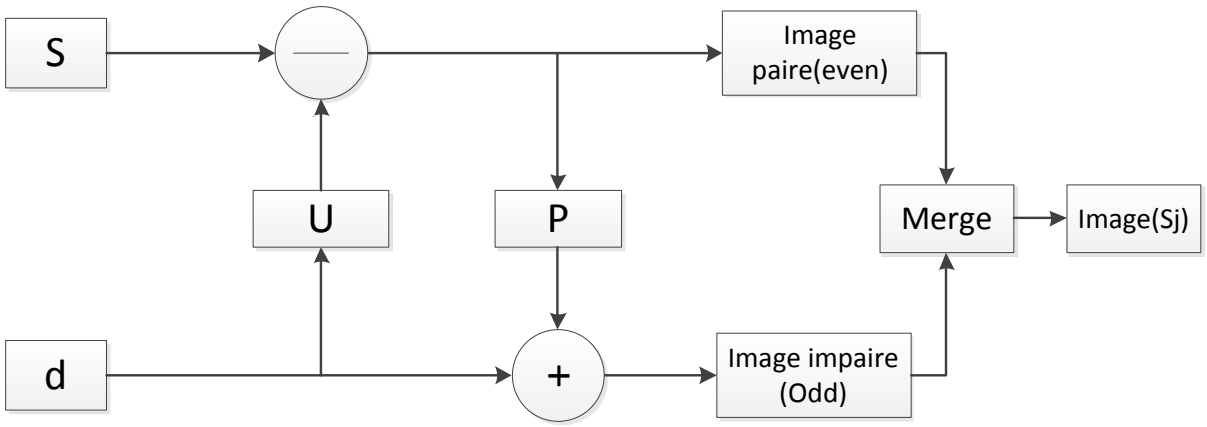

Figure 3.2: Schéma de L'algorithme du Lifting schème de la synthèse.

3.5.Avantages de l'Approche de *Lifting schème*

- Le *Lifting* permet une transformation "in place", allégeant de l'implémentation pour la transformée en ondelettes rapides. Ceci dit, pas besoin d'allouer un espace mémoire auxiliaire.

- L'usage du lifting permet particulièrement la construction des transformée en ondelettes non linéaires: on peut par exemple faire les transformées entier- entiers. Ceci est important pour les implémentations matérielles et pour le codage d'image.

- Toute transformation faite avec le lifting est immédiatement inversible et la transformée inverse a exactement le même coût en complexité, que la transformée elle-même.

- Le lifting permet une adaptation de la transformée en ondelette.

- Le lifting permet une construction des ondelettes sans faire usage de la transformée de Fourrier. Ceci signifie qu'il peut être utilisé pour construire les ondelettes qui ne sont pas nécessairement translatées ou dilatées d'une fonction (on parle d'ondelettes de seconde génération).

- Le lifting est plus souple pour les non puristes mathématiciens, car ne fait pas appel aux notions de la transformée de Fourrier, et peut être de la sorte facilement introduite, uniquement par ses arguments dans le domaine spatial.

- Enfin, le lifting expose le parallélisme inhérent de la transformée en ondelettes. Toutes les opérations d'un pas de lifting peuvent être faites totalement en parallèle.[JAN 15]

3.6.L'Application de *Lifting schème* sur les images.

Afin d'appliquer la transformée *Lifting schème* aux images, on procède simplement au découpage de l'image I(i,j) en deux images. La première image I even contient les éléments d'ordre pair (i + j est pair) de l'image de départ, la seconde I odd contient les éléments d'ordre impair (i + j est impair).

Ensuite, on applique la transformée sur les lignes (j) puis sur les colonnes (i) des images à l'aide des équations 3.1 et 3.2. Le résultat contient donc l'image d'approximation et les images de détails (horizontaux, verticaux et diagonaux) mélangées qu'il convient de remettre en ordre avant de les exploiter. [LAN 05]

		i						
		0	1	2	3	4	5	6
	0	e	o	e	o	e	o	e
	1	o	e	o	e	o	e	o
j	2	e	o	e	o	e	o	e
	3	o	e	o	e	o	e	o
	4	e	o	e	o	e	o	e

Figure 3.3: Les positions des pixels dans une image.

Le schéma suivant explique les résultats du traitement sur les lignes et colonnes.

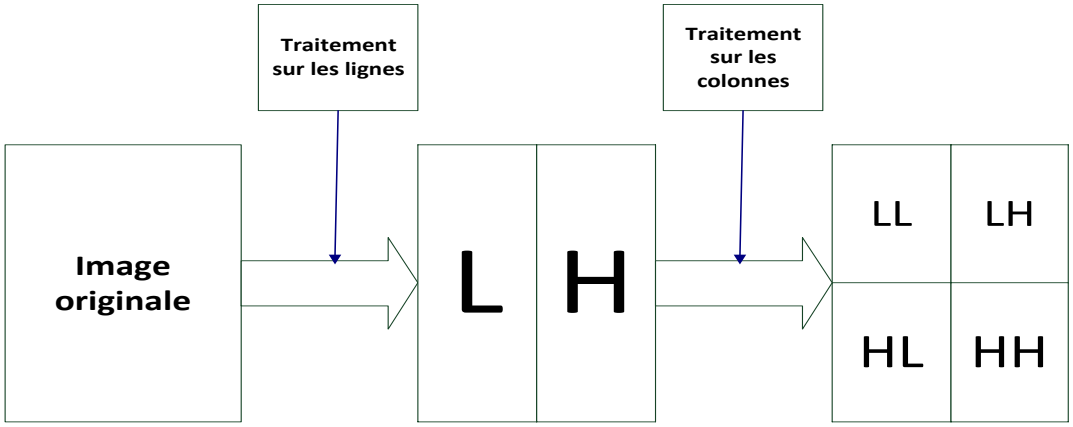

Figure 3.4: Décomposition en Ondelettes d'une image au premier niveau de résolution.

Pour le deuxième niveau de décomposition par *Lifting*, on applique les mêmes étapes précédentes sur l'image d'approximation (LL).

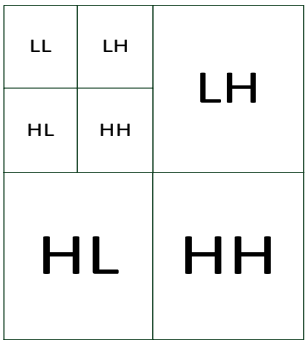

Figure 3.5: Image finale en deuxième niveau de résolution.

1) LL: image de basse fréquence.

2) LH: Il contient les informations horizontales de l'image.

3) HL: Il contient les informations verticales de l'image.

4) HH: Il contient les informations diagonales de l'image.

3.7.L'ondelette de *Haar* avec l'approche de *Lifting schème* :

La transformée de *Haar* est l'un des premières et les plus basiques transformées d'ondelette, dont la particularité est simple d'emploi pour le calcul. Elle est efficace du point de vue de consommation de mémoire, est réversible, rapide et simple.

Cette ondelette est très utilisée à nos jours. Les deux principales étapes de cet algorithme sont le calcul de l'addition et la soustraction. Ces deux opération qui généré les images de sous bandes (HH, HL et LH). [SAN 14]

Addition: $\frac{X(i) + X(i+1)}{2}$ *eq3.07*

La soustraction: $\frac{X(i) - X(i+1)}{2}$ *eq3.08*

3.7.1. Chronogramme de l'algorithme

Input: im:*image* , m:*nombre de lignes*, n:*nombre de colonnes*

Output: $img1$:*image approximation* , $img2$:*image detail*

For i = 1 **to** m **do** # i incrementer par 2

$img1 \leftarrow \frac{im(:,\ i) + im(:,\ i+1)}{2}$

end for

For i = 1 **to** m **do** # i incrementer par 2

$img2 \leftarrow \frac{im(:,\ i) - im(:,\ i+1)}{2}$

end for

Cette étape de l'algorithme est consacrée au traitement par les lignes, le résultat obtenu après cette étape est une image divisée sur deux.

input: $im1$:*image traité sur les lignes* , m:*nombre de lignes*, n:*nombre de colonnes*

Output: $img3$:*image approximation* , $img4$:*image detail*

For i = 1 **to** n **do** # *i incrementer par* 2

$$img3 \leftarrow \frac{im(i,:) + im(i+1,:)}{2}$$

end for

For i = 1 **to** n **do** # *i incrementer par* 2

$$img4 \leftarrow \frac{m(i,:) - m(i+1,:)}{2}$$

end for

Après un même travail est refait sur les colonnes et alors les 4 images (LL,LH,HL,HH) sont obtenues.

3.7.1. Simulation sur MATLAB

Le résultat de décomposition par la transformer de *Haar* avec l'approche de *Lifting schème* sur le simulateur MATLAB est présenté comme suit :

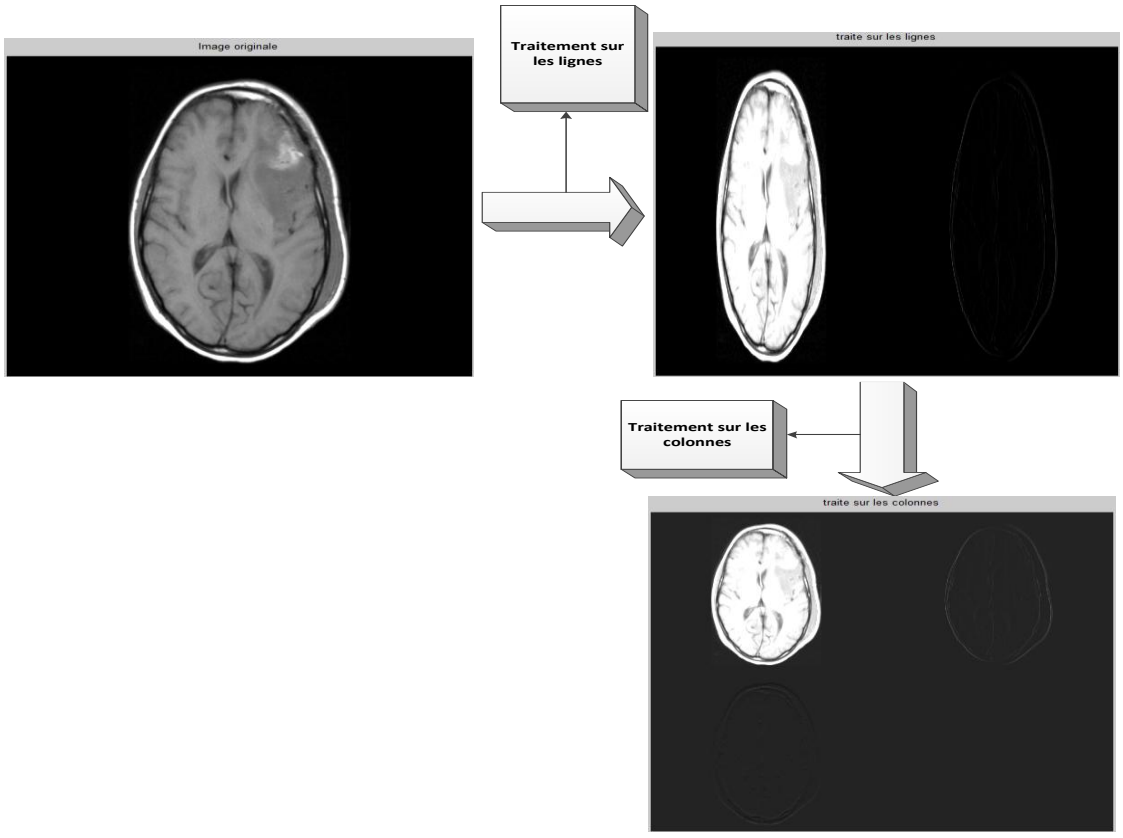

Figure 3.6: Résultats de simulation sur MATLAB

3.8. Algorithme *lifting schème* généralisé :

3.8.1. Chronogramme de l'algorithme :

On décompose l'image en deux images : paire et impaire.

input: m:image , Il:nombre de lignes, Ic:nombre de colonnes

Output: $img1$:image paire , $img2$:image impaire

For $i = 1$ ***to*** Il ***do***

For $j = 1$ ***to*** Ic ***do***

$b_n \leftarrow i + j$

IF $mod(b_n) \leftarrow 0$ ***then***

$img1 \leftarrow m\,(i,j)$

else

$img2 \leftarrow m\,(i,j)$

end if

end for

end for

Ensuite on utilise ces deux images pour faire le traitement sur les lignes.

input: $img1$:*image paire* , $img2$:*image impaire* , Il:*nombre de lignes*, Ic:*nombre de colonnes*,

output: img_L:*image d'approximation*, img_H: *image de détail*

For $i = 1$ ***to*** Il ***do***

For $j = 1$ ***to*** Ic ***do***

$img_H \leftarrow img2 - \frac{1}{2} img1$

end for

end for

For $i = 1$ ***to*** Il ***do***

For $j = 1$ ***to*** $\mathbf{Ic}$ ***do***

$img_L \leftarrow img1 + \frac{1}{2} img_H$

end for

end for

Le même traitement sera fait sur les colonnes, on prend img_L et img_H comme entrée. À la fin, on aura 4 images résultantes img_{LL} et img_{LH} et img_{HL} etimg_{HH}.

3.8.2. Simulation sur MATLAB

Le résultat de décomposition sur le simulateur MATLAB est illustré sur la figure 3.1 :

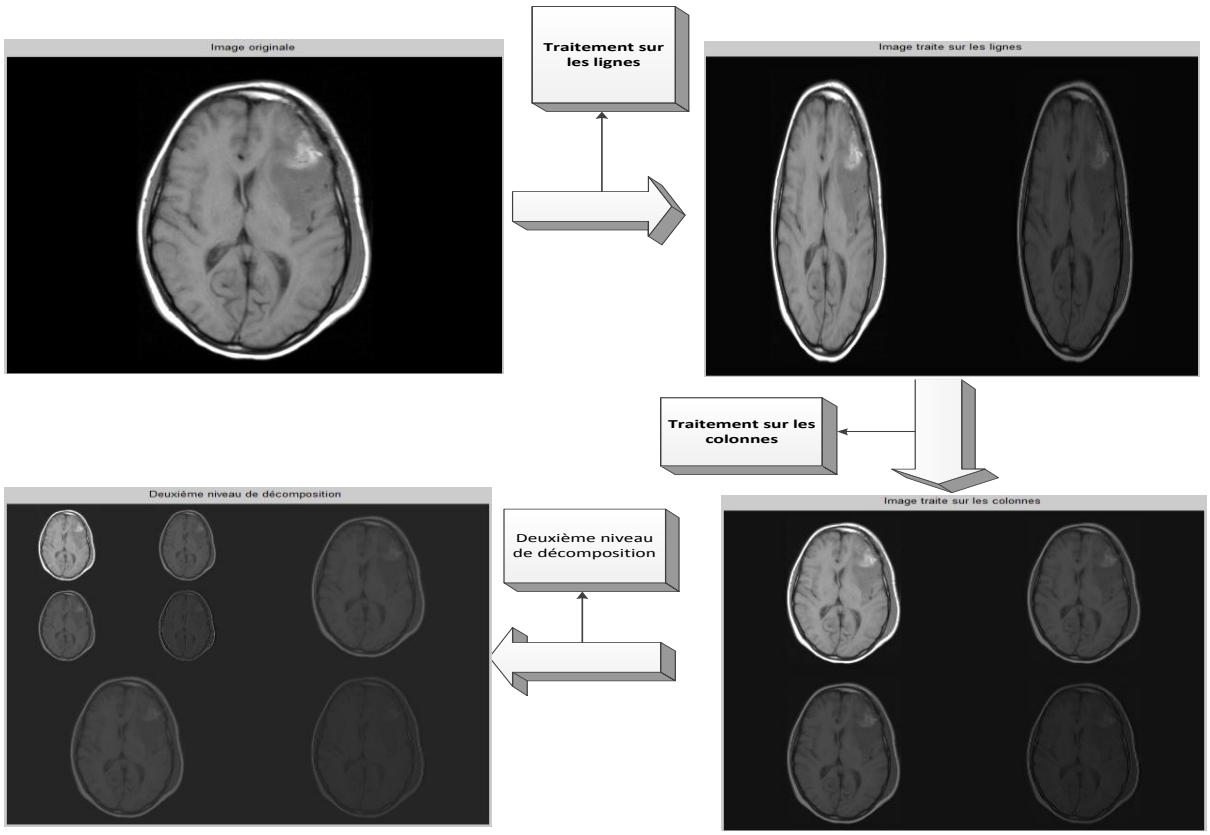

Figure 3.7 : Deuxième niveau de décomposition par Lifting schème d'une image médicale.

3.8.3. Algorithme de *Lifting Scheme* pour la transformée en ondelette de *Daubechies* D4 :

Dans cette partie, une ondelette plus complexe avec quatre coefficients est utilisée :

($\alpha = -\sqrt{3}$ $\Upsilon = \frac{\sqrt{3}}{4}$ $\beta = \frac{\sqrt{3}-2}{4}$ $\delta = \frac{\sqrt{3}+1}{\sqrt{2}}$) .Cette partie montre la façon dont est appliquée les étapes de *Lifting scheme* sur l'ondelette de ***Daubechies*** D4 comme le montre la figure suivante :

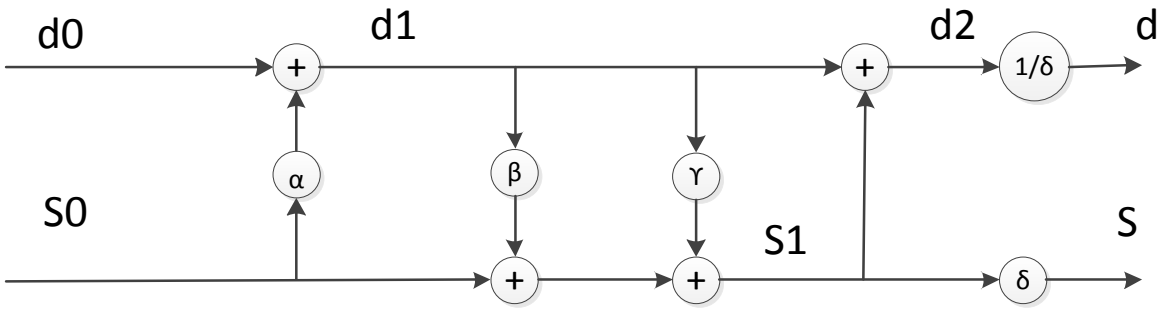

*Figure 3.8: Schéma de l'implémentation de l'ondelette de **Daubechies**.*

Algorithme de *Lifting schème* pour l'ondelette de *Daubechies*

input: m:image , Il:nombre de lignes, Ic:nombre de colonnes

Output: S0:image paire , d0:image impaire

For i = 1 **to** Il **do**

For j = 1 **to** Ic **do**

$b_n \leftarrow i + j$

IF $\mathrm{mod}(b_n) \leftarrow 0$ **then**

$S0 \leftarrow m(i,j)$

else

$d0 \leftarrow m(i,j)$

end if

end for

end for

Traitement sur les lignes

input: S0:image paire , d0 : image impaire Il:nombre de lignes, Ic:nombre de colonnes,

output: S:image d'approximation, d: image de détail

For i = 1 **to** Il **do**

For j = 1 **to** Ic **do**

$d1 \leftarrow d0 - \sqrt{3} * S0$

end for

end for

For i = 1 **to** Il **do**

For j = 1 **to** Ic **do**

$S1 \leftarrow S0 + \frac{\sqrt{3}}{4} * d1 + \frac{\sqrt{3}-2}{4} * d1$

end for

end for

For i = 1 **to** Il **do**

For j = 1 **to** Ic **do**

$d2 \leftarrow d1 + S1$

end for

end for

For i = 1 **to** Il **do**

For j = 1 **to** Ic **do**

$S \leftarrow \frac{\sqrt{3}+1}{\sqrt{2}} * S1$

end for

end for

For i = 1 **to** Il **do**

For j = 1 **to** Ic **do**

$d \leftarrow \frac{1}{\frac{\sqrt{3}+1}{\sqrt{2}}} * d2$

end for

end for

Le même algorithme sera appliqué sur les colonnes pour obtenir les quatre images.

3.9.Conclusion :

Dans ce chapitre, nous avons introduit les deux mécanismes de décomposition d'ondelette de ***Haar*** par *Lifting schème*. Le second algorithme de décomposition généralisé présente une meilleurs performance pour mettre en œuvre les transformées en ondelettes.

Dans le dernier chapitre nous allons proposer l'implémentation hardware des deux algorithmes.

4. Chapitre 4 : Implémentation

4.1. Introduction

L'objectif principal de ce chapitre est l'implémentation matérielle (hardware) sur un circuit FPGA (Field Programmable Gate Array) d'une architecture performante de la méthode des bandelettes sur une plate-forme ZedBoard de la famille Zynq.

Dans un premier temps, deux architecture de l'ondelette de ***Haar*** ont été développées et simulées sous trois environnements de développement, à savoir : Xilinx ISE 14.7 et Vivado2015.4 et SDK 2015.4.

Dans ce chapitre nous avons commencé par la présentation de la méthodologie de l'implémentation de l'algorithme de bandelette par l'implémentation des deux architectures :

- Le premier basé sur l'approche de lifting schème avec *Haar*,
- Le second basé sur le *Lifting schème* généralisé,
- Et description du co-design du système global.

4.2. L'architecture globale de la méthode de transformée en Bandelettes:

L'algorithme de bandelette s'avère long et compliqué pour l'implémenter, donc il est nécessaire de choisir un système embarqué pour la réalisation de ce système. Ce qui fait, ce travail est divisé en deux parties, l'une est de type software et l'autre est de type hardware comme le montre la figure 4.1:

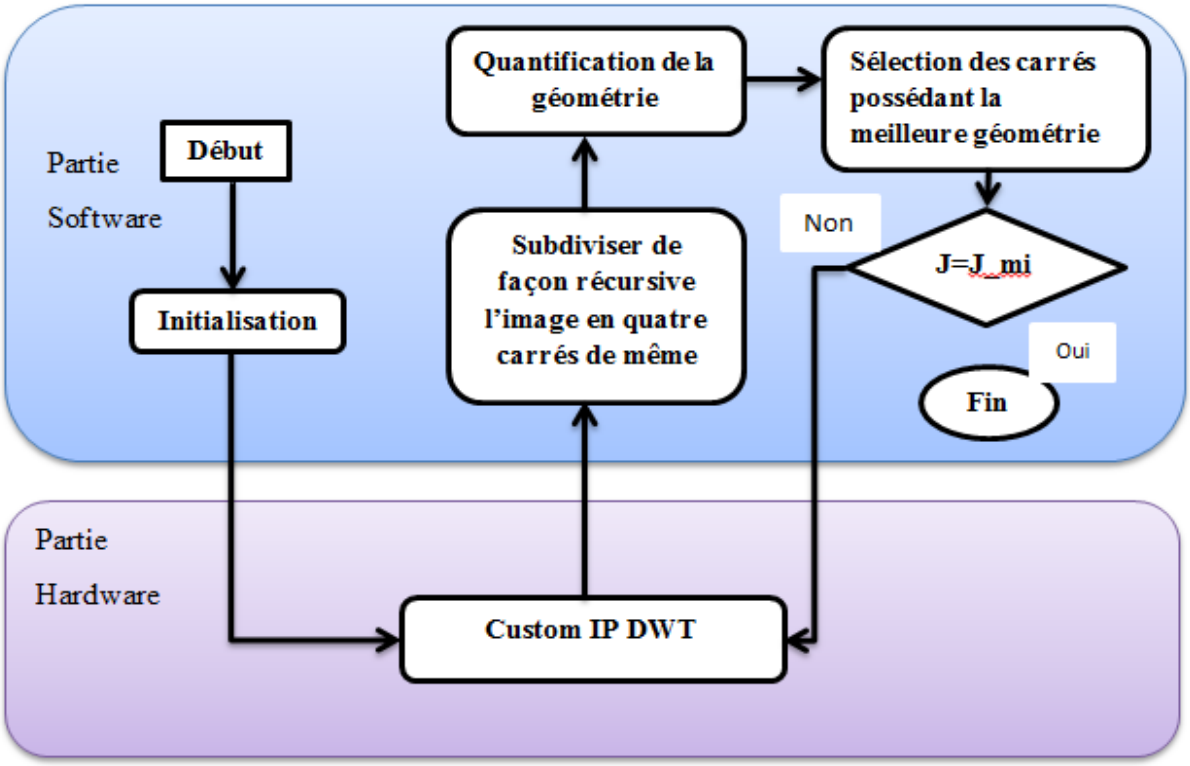

Figure 4.1 *: Schéma Synoptique des différentes parties de l'algorithme de bandelette.*

On a choisi la partie qui subdivise de façon récursive l'image en quatre carrés de même taille et la partie qui quantifie la géométrie et sélectionne les carrée qui possèdent la meilleure géométrie, qui est considéré comme complexe car il faut l'exécuter instruction par instruction ; ce qui représente la partie software PS qui a été implémenté sur le microprocesseur ARM cortex_9.

Ainsi nous avons implémenté la premier partie de l'algorithme de la décomposition en ondelette le *Lifting schème* dans la partie hard PL implémentée sur FPGA donc il existe une possibilité pour le parallélisme de cette partie.

4.3. Description de l'architecture de l'ondelette de *Haar* par *Lifting schème*

L'architecture développée peut être résumée en deux grandes parties : la première est une architecture dédiée au traitement des lignes et la deuxième représente le traitement des colonnes. Le schéma synoptique des deux architectures est représenté dans la figure ci-dessous:

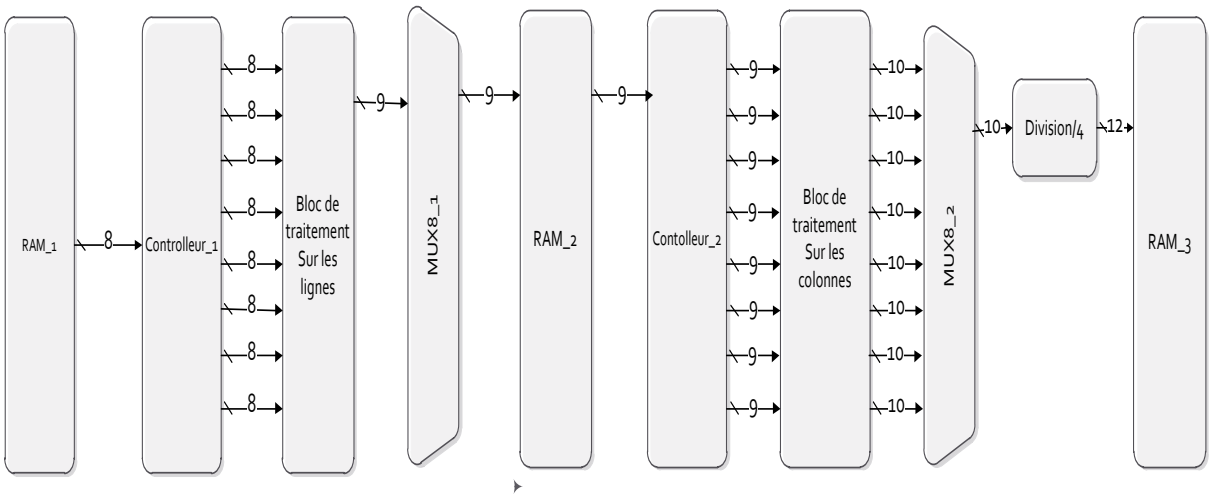

Figure 4.2: Schéma synoptique l'ondelette de Haar.

4.3.1. Traitement sur les lignes.

L'exécution du traitement des lignes de l'image se fait par les différents blocs suivants :

- **Stockage de l'image originale dans la RAM_1**

L'image va subir de nombreuses transformations. Au début elle va être enregistrée sous la forme d'une matrice de pixels, ensuite cette dernières va être convertie en

vecteur à travers un programme software. L'image sous sa forme finale (vecteur) (figure 4.3) sera stockée ligne par ligne dans une RAM appelée RAM_1.

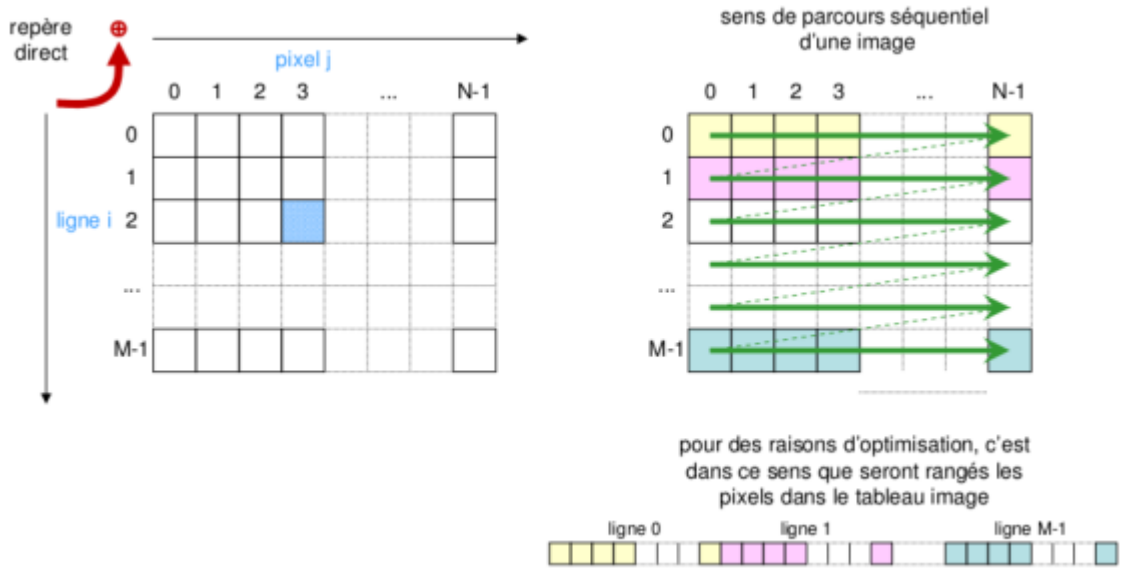

Figure 4.3: Le stockage de l'image dans la RAM.

Le stockage dans la RAM noté RAM_1. Des pixels commencent par l'émission du contrôleur de deux signaux, le premier est un signal d'autorisation de lecture RW_RAM_1 =1 pour la RAM_1. Et le deuxième est un signal d'adressage (addr_RAM_1) qui définit la position des pixels dans la RAM, dans notre cas, ce signal va être incrémenté de 1 jusqu'à 64.

- **Lecture des pixels de la RAM_1**

Après le stockage de tout les Pixels (64 pixels) sur la RAM_1, Le contrôleur envoie deux signaux avec deux messages : (lecture et adressage) (voir dans l'annexe 02 le programme VHDL de la RAM.

Signal de lecture RW_RAM_1=0 : définit l'ordre d'envoi des pixels concernés à la sortie de la RAM vers le contrôleur**.**

Signal de l'adressage addr_RAM_1: définit la position du pixel concerné.

Le contrôleur reçoit un pixel pour chaque front montant d'horloge (clk), ce dernier est stocké dans un latch, pour être transmis par la suite dans le bloc de traitement. Pour faire des opérations arithmétiques et permettre de réaliser des soustractions et des additions pour chaque deux pixels adjacents : {1 , 2} , {3 , 4},…… {7 , 8}.

Afin d'optimiser l'opération, le contrôleur procède à l'envoi de 16 pixels (8 pixels envoyés en deux fois) 8 pour la soustraction (PP0-PP1…PP6-PP7) et 8 pour l'addition (P0+P1...P6+P7). Résultats présentés dans la figure 4.4.

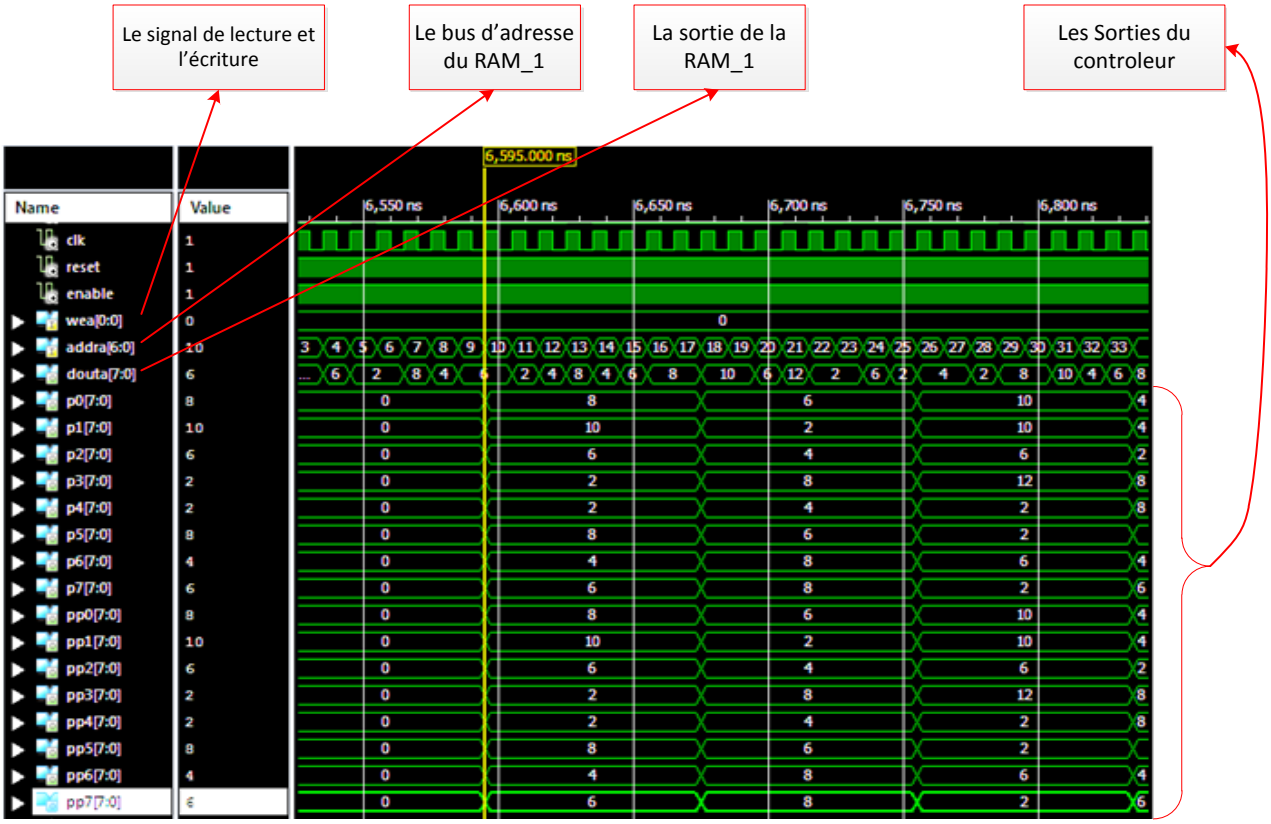

Figure 4.4: Simulation de la mémoire RAM_1 et le contrôleur_1.

- **Bloc de traitement sur les lignes**

Le schéma de la figure 4.2 contient 4 additionneurs et 4 soustracteurs afin d'effectuer l'addition et la soustraction de chaque deux pixels adjacents (P0+P1, P2+P3, P4+P5, P6+P7) et (PP0-PP1, PP2-PP3, PP4-PP5, PP6-PP7) à la fois. Les résultats sont représentés sur la figure (4.5) par les signaux (out_adder0,… out_adder3 et out_substract0, … out_substract3).

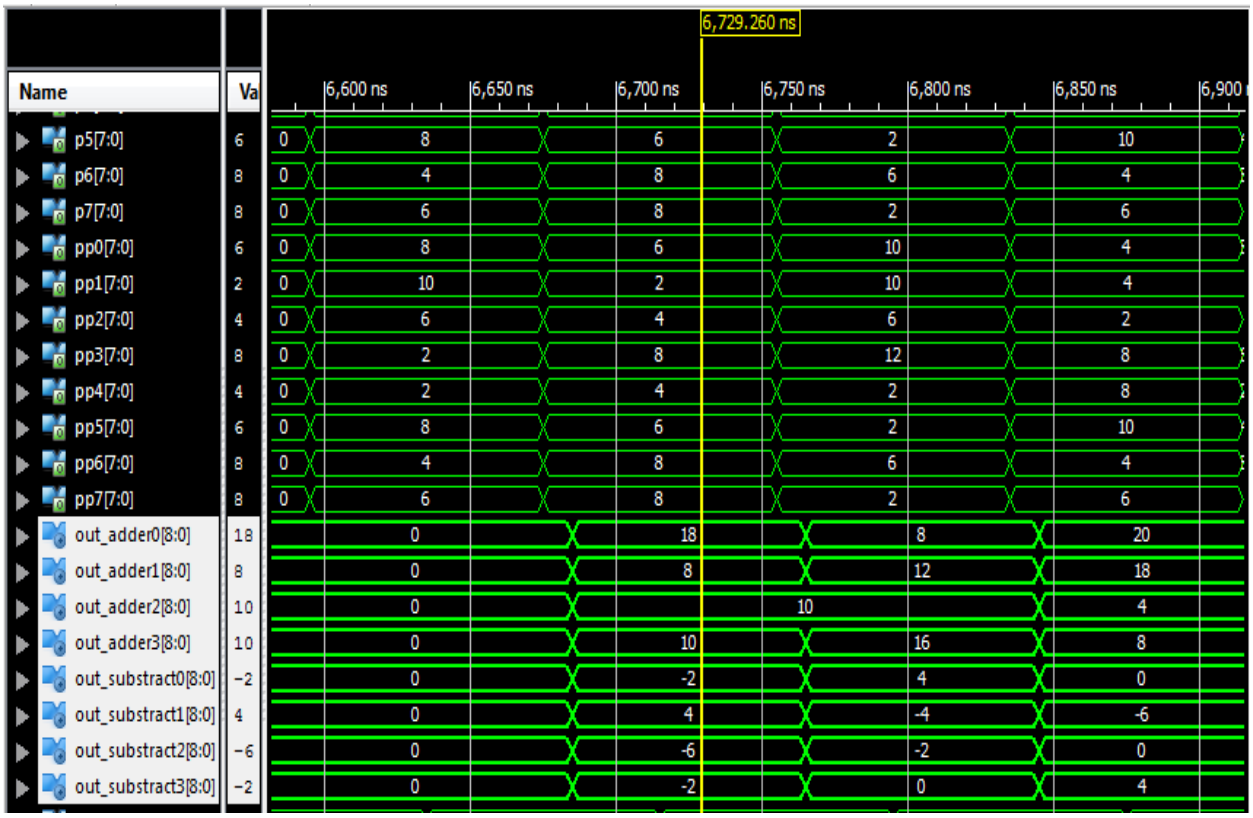

Figure 4.5: Simulation du bloc de traitement.

La sorties du bloc figure 4.2 de traitement sur les lignes est reliée à l'entrée du multiplexeur (MUX 8 vers 1). Le contrôleur nous donne un signal de sélection du multiplexeur pour choisir le pixel à la sortie du multiplexeur pour faciliter le stockage des pixels traités dans la RAM_2.

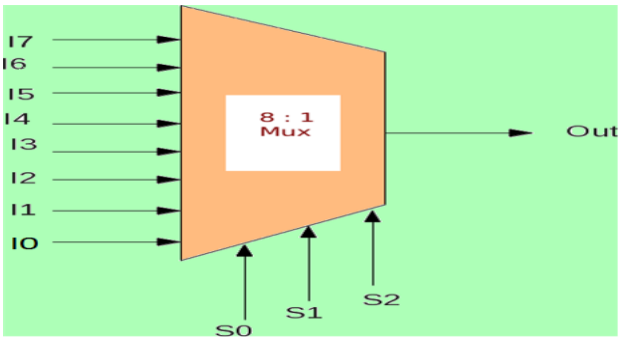

Figure 4.6 :MUX_8 vers 1.

- **Stockage des pixels traités dans la RAM_2.**

Le stockage est fait lorsque le contrôleur envoie un signal de l'écriture (RW_RAM_2=1) vers la RAM_2 et un autre signal de l'adressage (addr_RAM_2) qui s'incrémente par 1 (1,2,…,64) et la sortie du multiplexeur est reliée à l'entrée de la RAM_2. Les résultats de simulation sont montrés sur la figure 4.7.

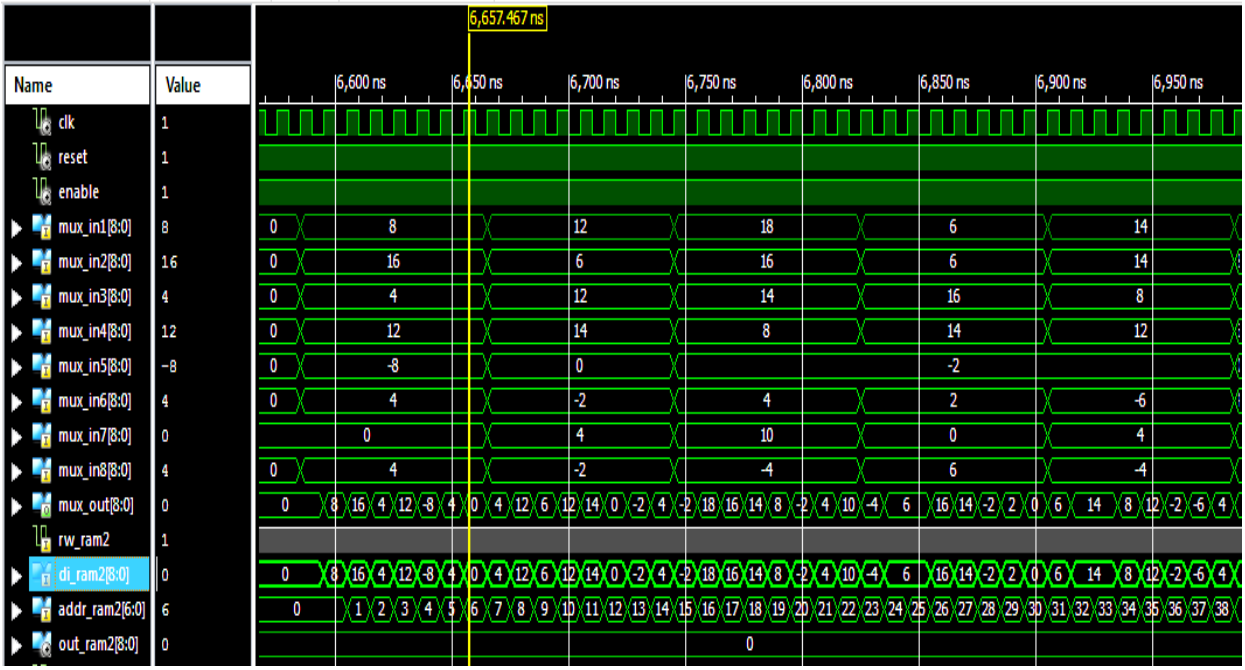

Figure 4.7: Simulation du MUX_2 et RAM_2.

Une fois le traitement des lignes terminé, nous obtenons l'image divisée en deux.

4.3.2. Traitement sur les colonnes

Les mêmes étapes sont effectuées pour les colonnes.

Lectures des données traitées sur les lignes par le controleur_2

Ce contrôleur _2 a le même rôle que le contrôleur_1 : il envoie un signal de lecture RW_RAM2=0, mais cette fois-ci un changement s'opère dans l'adressage de la lecture de la RAM_2, qui effectue une incrémentation par 8 (0, 8,16…56,1,9…64) dans le but est de prendre les pixels de chaque colonne (out_ram2) afin d'obtenir à la sortie 16 pixels (8 et 8 identique) orientés vers le bloc de traitement_2 comme le montre la simulation de figure 4.8.

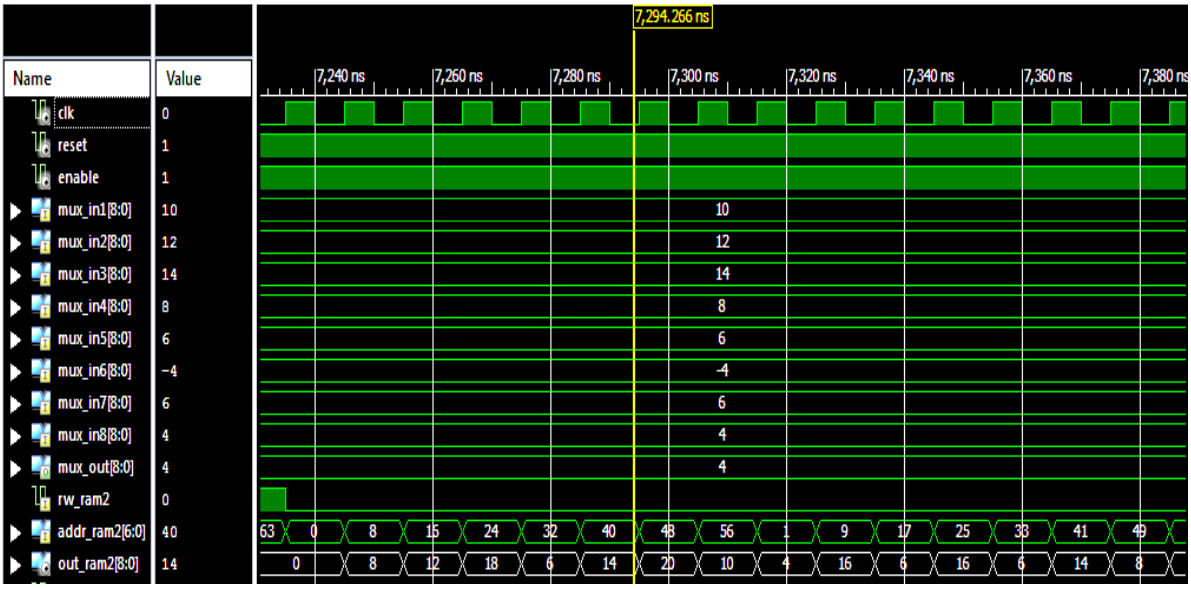

Figure 4.8: Simulation de la sortie de la RAM_2.

- **Bloc de traitement_2**

Ce bloc contient 4 additionneurs et 4 soustracteurs pour l'addition et la soustraction de deux pixels adjacents, la sortie est reliée au multiplexeur MUX_2 (MUX 8 :1). La sortie de MUX_2 est adaptée à l'entrée du diviseur qui effectue une division de la donnée sur 4. Cette division est accomplie avec un décalage de deux bits à droite. Ensuite, les données sont stockées dans une RAM_3 après l'émission du controleur_2 d'un signal RW_RAM_3=1 pour l'écriture sur la RAM_3 et un autre signal de l'adressage addr_ram3 comme il est illustré dans la figure 4.9.

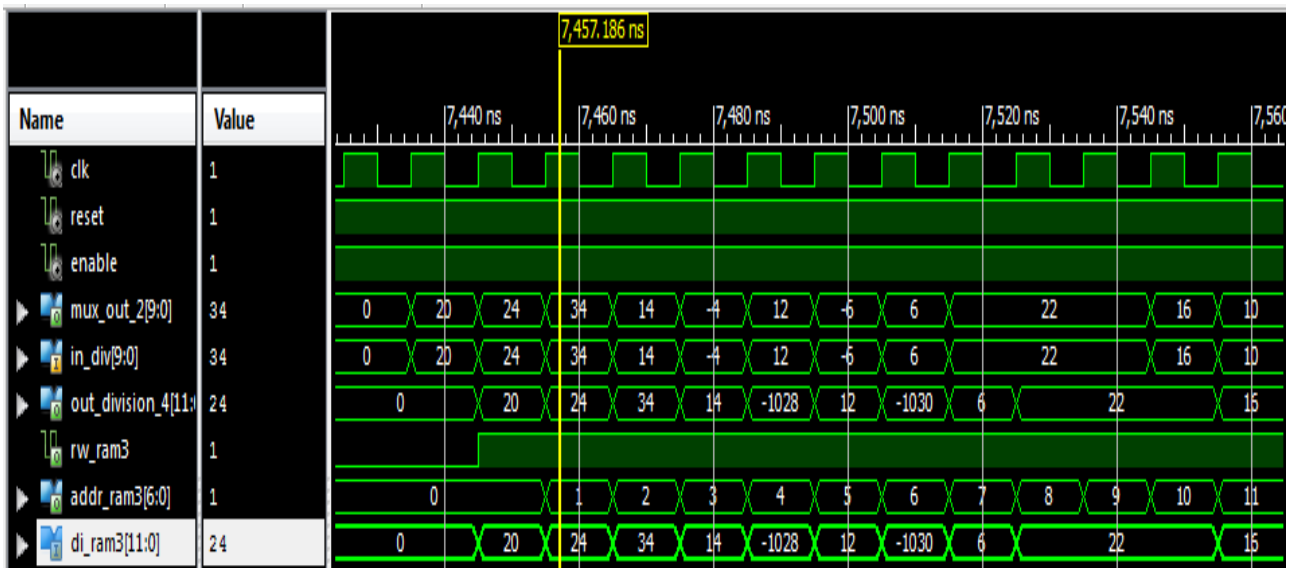

Figure 4.9:Simulation de sortie du MUX_2, de la division, et de la RAM_3.

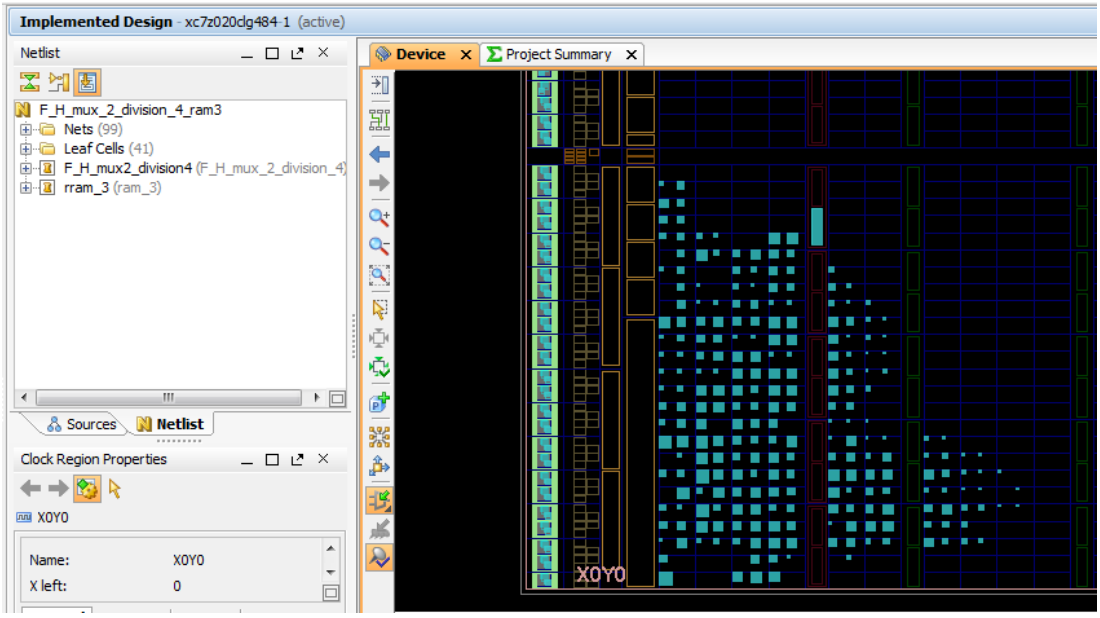

Figure 4.10: La surface occupée par l'architecture de Lifting scheme.

4.4. Description de l'architecture de l'ondelette par *Lifting scheme* généralisé.

Le schéma synoptique du l'architecture l'ondelette de *Haar* par la méthode de *Lifting schème* généralisé est présenté dans la figure 4.11.

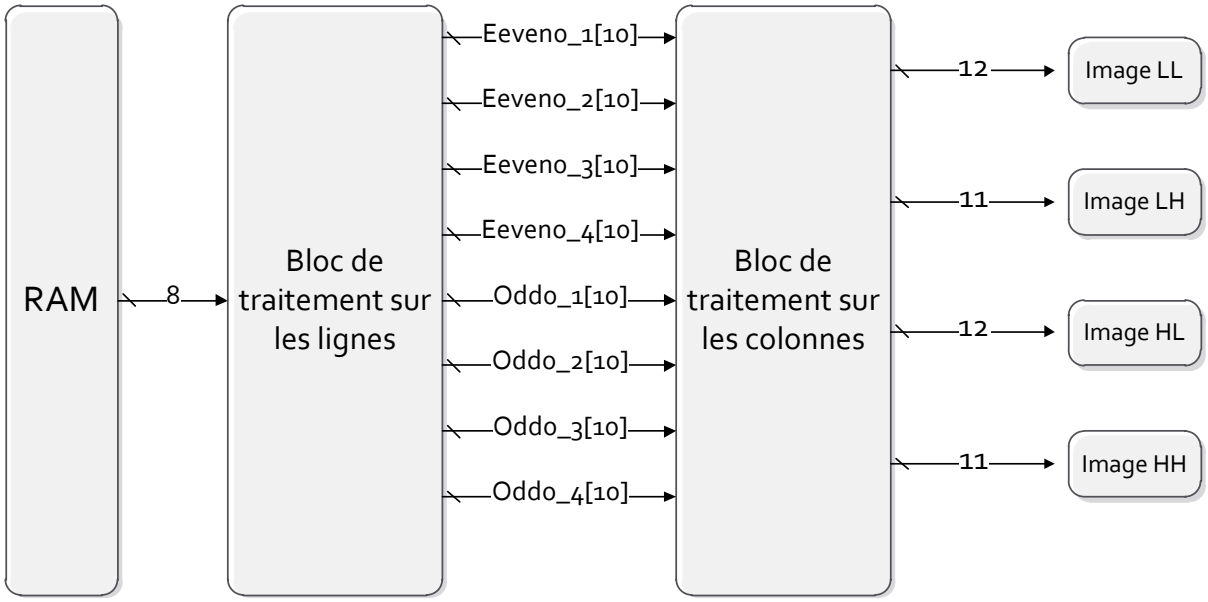

Figure 4.11:L'architecture globale de Lifting scheme.

4.4.1. Stockage de l'image originale dans la RAM_1

Dans cette architecture, une partie de l'image originale (64 pixels) est stockée après un traitement software qui effectue la division des positions des pixels dans l'image originale en deux vecteurs contenant les pixels occupant les positions paires « even » et les positions impaires « odd » (*dans la matrice des pixels de l'image originale*).

Matrice M

11	3	7	4	18	12	8	4
8	4	32	0	10	2	15	8
9	0	10	2	13	4	6	33
16	4	6	0	14	2	8	20
15	4	20	2	10	22	4	8
7	6	12	2	10	21	4	8
9	6	0	8	6	4	11	2
10	5	6	16	2	14	4	8

Matrice de l'image originale

Les pixels impairs

3 4 12 4 8 32 10 15 0 2
4 33 16 6 14 8 4 2 22 8
7 12 10 4 6 8 4 2 10 6
2 4

Les pixels pairs

11 7 18 8 4 0 2 8 9 10
13 6 4 0 2 20 15 20 10
4 6 2 21 8 9 0 6 11 5
16 14 8

Une fois la communication établie entre le contrôleur et la RAM, les pixels « odd » sont stockés après les pixels « even » dans la RAM par l'envoi du contrôleur un signal de l'écriture RW_RAM = 1 et un signal d'adressage addr_RAM qui incrémente par 1 (0.1.2.3…63). Les 32 premières cases mémoire contenant les pixels impairs « odd » et l'autre contenant les pixels pairs « even », comme c'est représenté dans la figure 4.12.

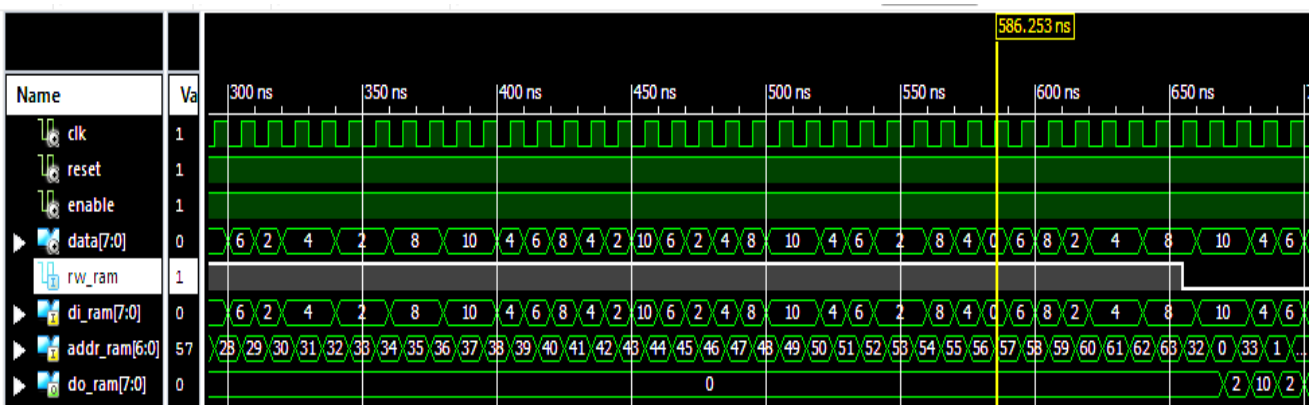

Figure 4.12:Simulation de l'écriture de la ram_1.

4.4.2. Traitement sur les lignes

Cette étape est élaborée après l'envoi du contrôleur un signal de lecture RW_RAM_1=0 et un signal de l'adressage addr_RAM_1 qui change afin de lire un pixel « even » puis un pixel « odd » (32.0.33.1…63.31). A chaque front montant de l'horloge le bloc de traitement reçoit un pixel. Chaque pixel est stocké dans un latch (p0_latch, p1_latch, p2_latch, p3_latch, p4_latch, p5_latch, p6_latch, p7_latch) afin de préparer les pixels à l'étape de prédiction qui fait le traitement de chaque deux pixels adjacents en synchronisant cette phase par une machine à états finis résultats, figure 4.13.

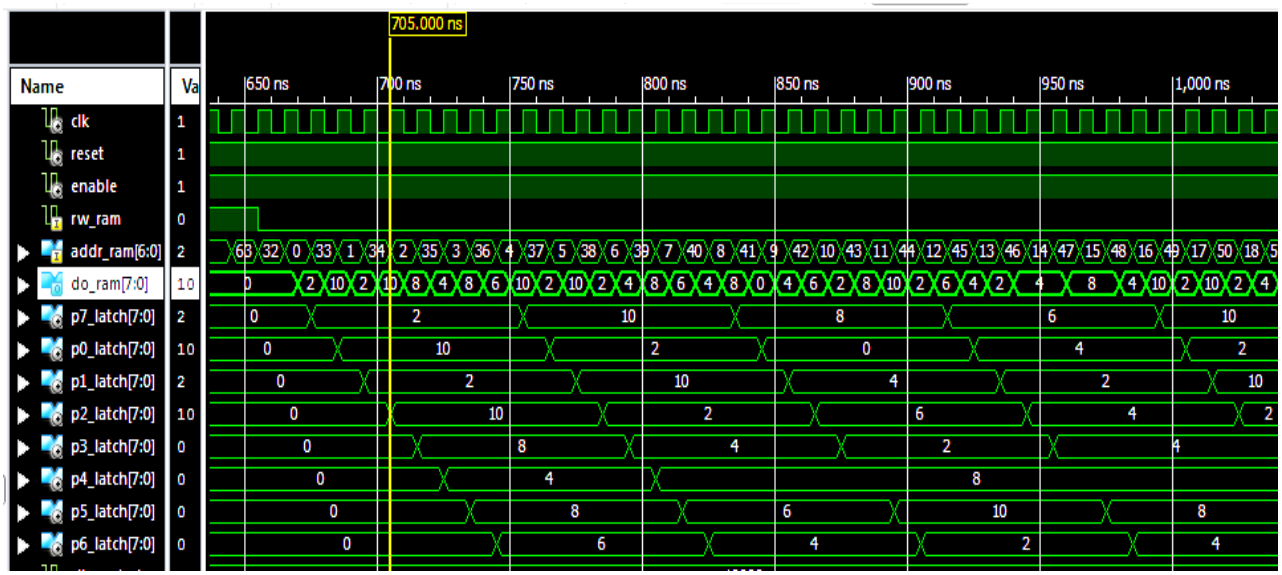

Figure 4.13: Simulation des pixels stockés dans les latchs.

- **Prédiction**

Une multiplication des pixels « even » par les coefficients d'approximation de l'ondelette (Haar =1/2) puis on divise par un décalage à droite d'un bit, après une soustraction par les pixels « odd » (eq3.1). Dans cette étape, nous trouvons les pixels haute fréquence de l'image.

- **Mise à jour (Update)**

Une multiplication de la sortie de prédiction par le coefficient de dilatation d'ondelette (Haar= 1/2) puis une addition avec le pixel « even » (latch). Dans cette étape, nous trouvons les pixels basse fréquence de l'image.

Exemple

Afin d'expliciter les deux étapes précédentes, nous proposons cet exemple : p0_latch représente le pixel pair « even » et p1_latch représente le pixel impair « odd », puis on divise p0_latch ; le résultat sera pris par un signal div_even0_4 ensuite une soustraction par p1_latch dans un signal sig_odd1_1 est effectuée, puis nous élaborons une division sig_odd1_1 pour trouver div_odd1_1, enfin nous accomplissons l'addition par p0_latch. Le code VHDL de cet exemple est donné en annexe 02.

.

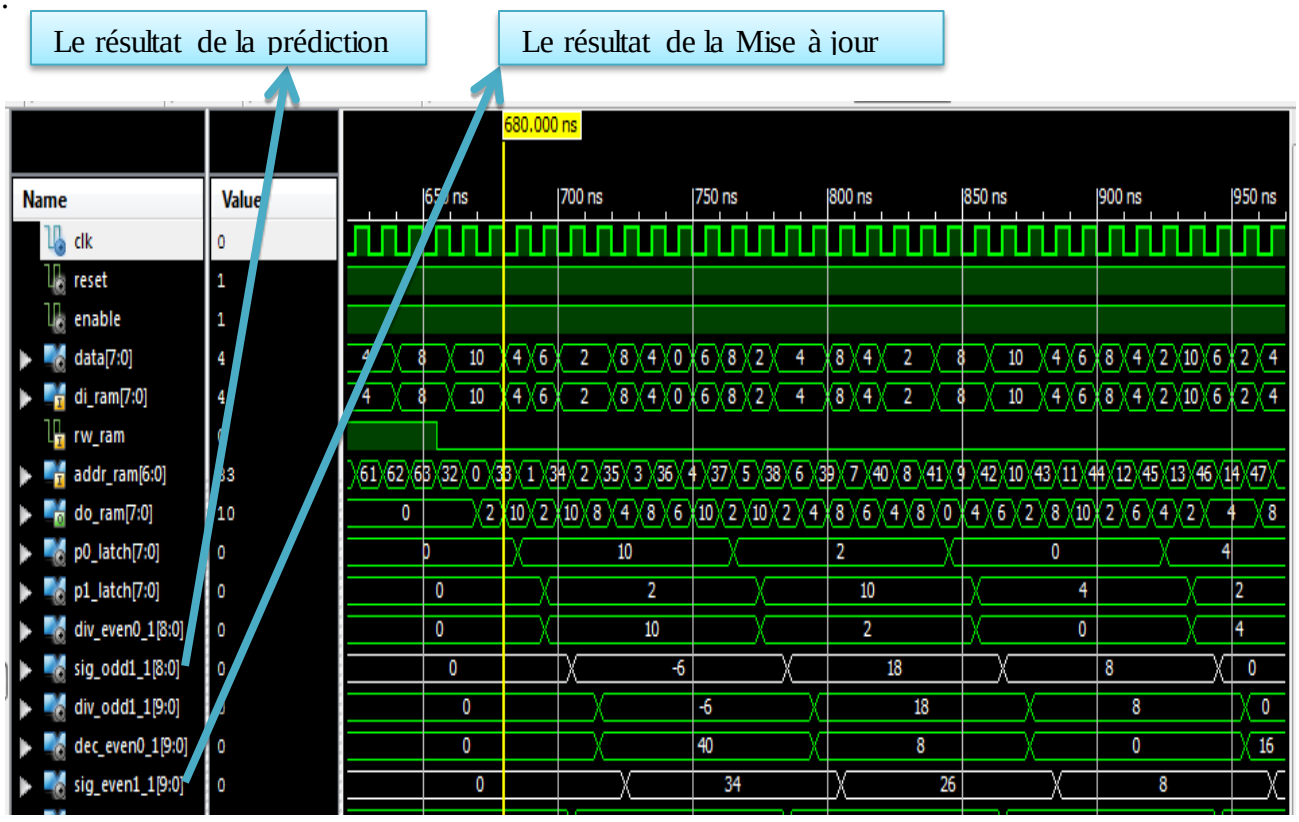

Figure 4.14: Simulation des deux étapes : Prédiction et Mise à jour.

Ces deux étapes sont répétées simultanément 4 fois afin d'accélérer le travail ; 4 signaux (even0_1…even0_4) sont retrouvés à la sortie du bloc de traitement

représentant les pixels basse fréquence et 4 autre signaux (odd0_1 …odd0_4) représentant les pixels haute fréquence.

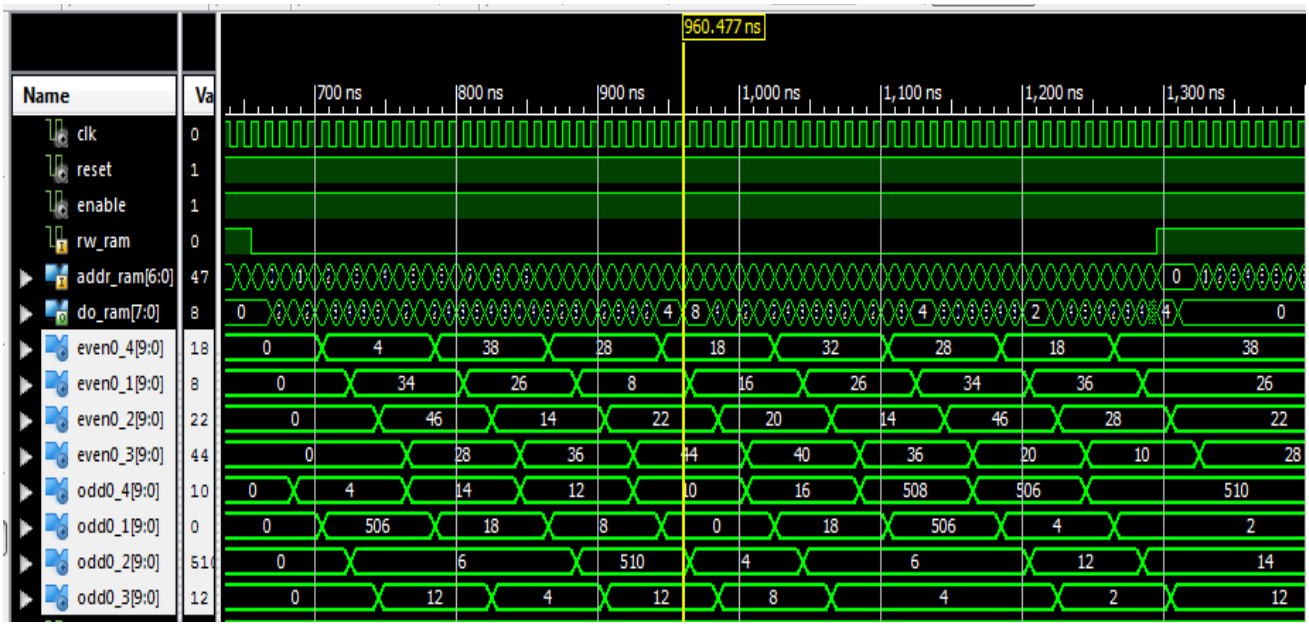

Figure 4.15: Simulation du résultat de bloc de traitement sur les lignes.

4.4.3. Traitement sur les colonnes

Le bloc de traitement sur les colonnes reçoit les sorties du bloc de traitement sur les lignes. Les signaux even0_1, even0_2, even0_3, even0_4 représentent l'image en basse fréquence et les signaux odd0_1 … odd0_4 représentent l'image en haute fréquence. Et afin de mettre en œuvre la séparation (split) sur les colonnes, il faut connaitre les positions des pixels (tableau 4.1).

Even0_1	E	O	E	O	E	O	E	O
Even0_2	O	E	O	E	O	E	O	E
Even0_3	E	O	E	O	E	O	E	O
Even0_4	O	E	O	E	O	E	O	E

Tableau 4.1 : Représentation des positions des pixels sur chaque signal.

Les pixels des signaux « « even » et « odd » seront séparés et stockés par type (pair et impair) dans des latchs différents pour le traitement.

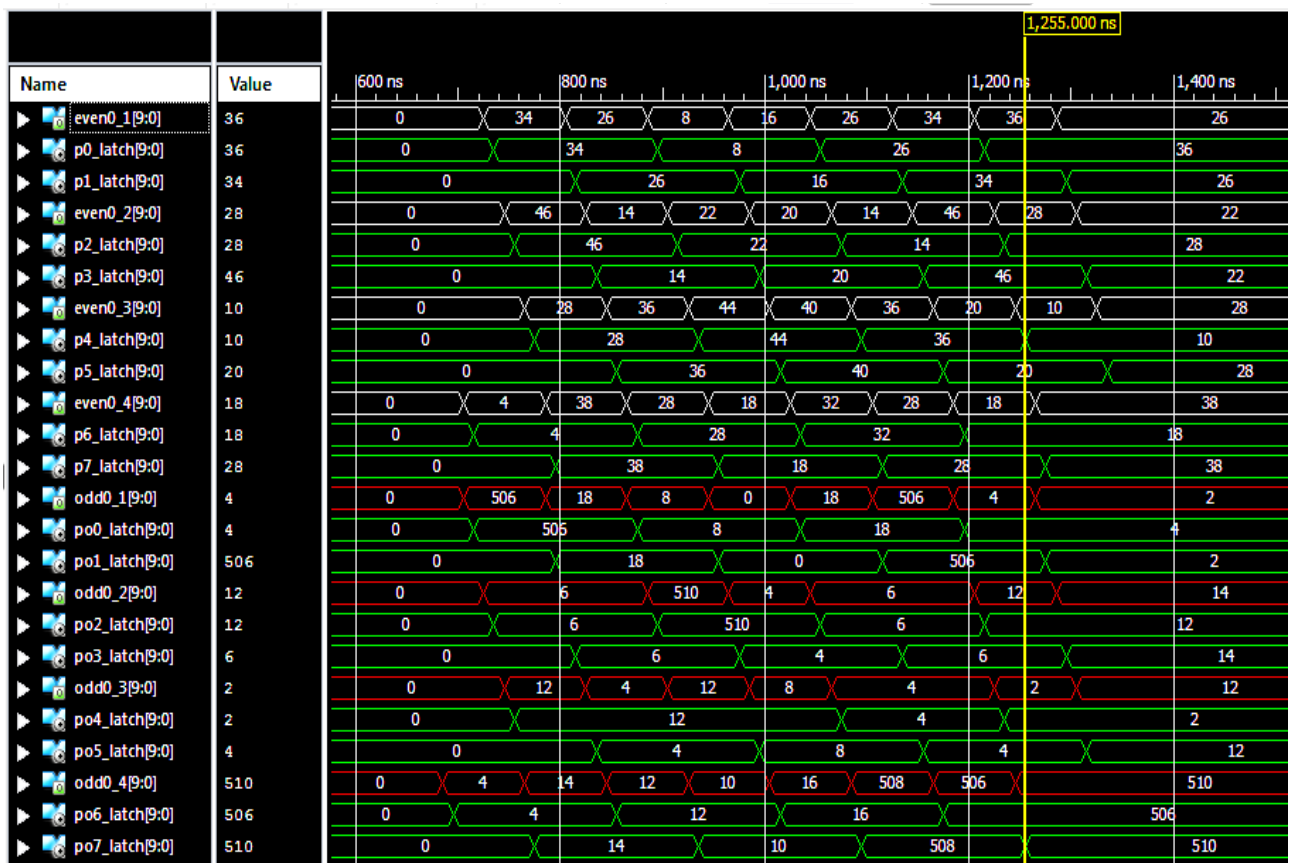

Figure 4.16: Simulation pour la séparation (split) des pixels.

- **Prédiction et mise à jour**

Le même traitement avec les mêmes coefficients de ***Haar*** est utilisé pour les colonnes.

Exemple le signal **even0_1** va être stocké dans les latchs **p0_latch** (pixels pairs) et **p1_latch** (pixels impairs) dans la figure 4.17.

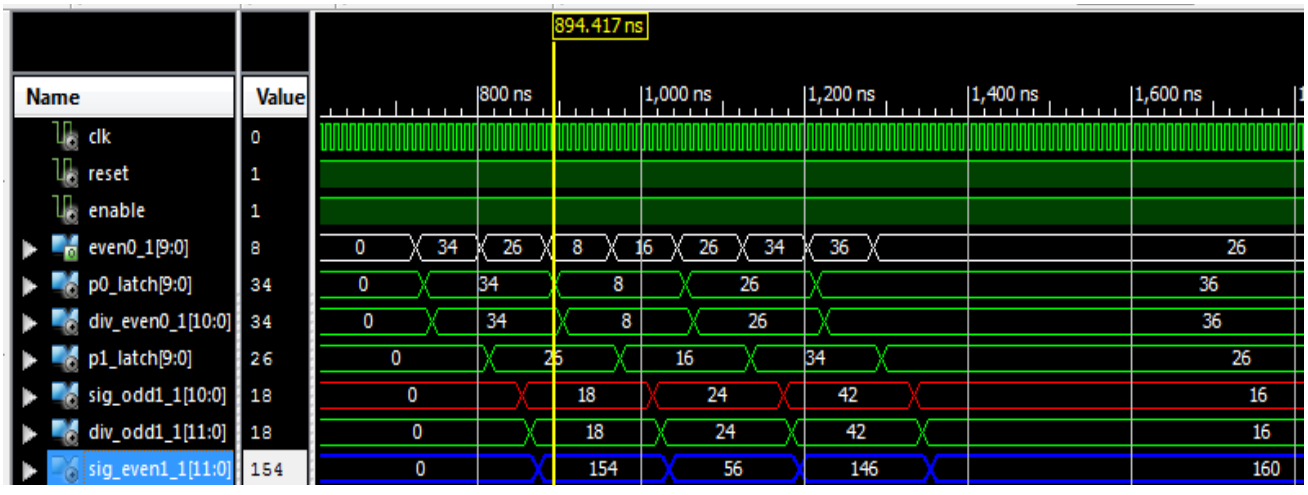

Figure 4.17:Simulation du traitement sur les colonnes de signal even0_1.

Après le traitement de chaque signal « even » le résultat est comme suit :

- sig_odd1_1 contient les informations horizontales de l'image (LH)
- sig_even1_1 contient des informations de l'image d'approximation (LL)

Pour les signaux « odd » :

- sigo_odd1_1 contient les informations diagonales de l'image (HH)
- sigo_even1_1 des informations verticales de l'image (HL)

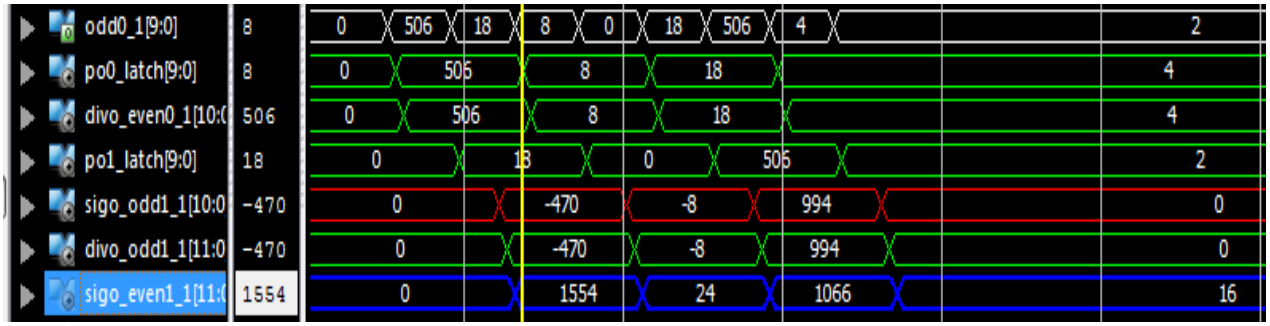

Figure 4.18: Simulation du traitement sur les colonnes du signal odd0_1.

Après le rassemblement de tous les signaux, le résultat final sera comme suit :

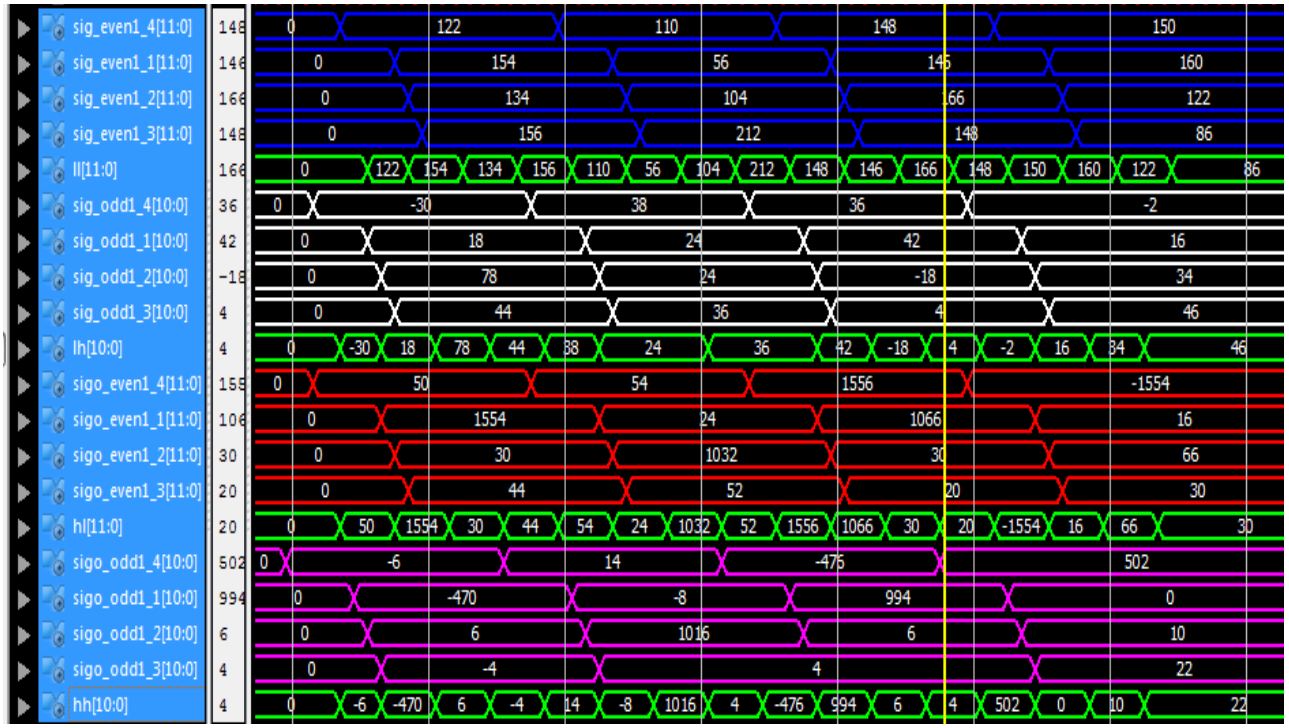

Figure 4.19: Simulation des signaux qui contiennent la même information.

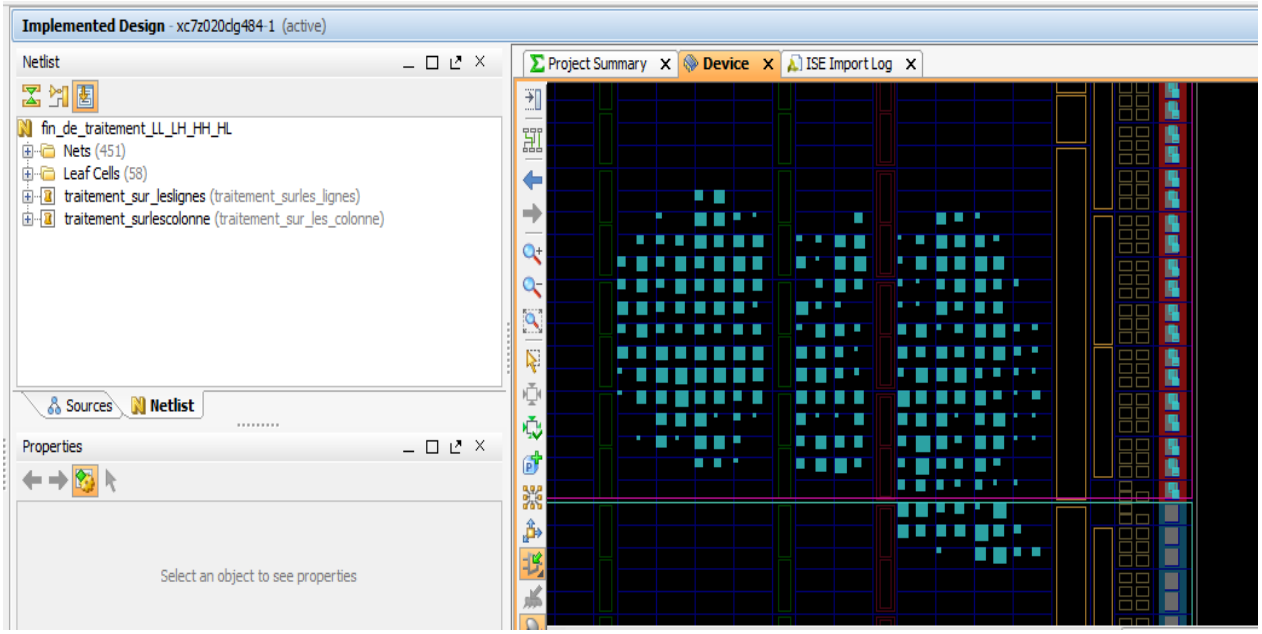

Figure 4.20: La surface occupée par l'architecture de Lifting scheme généralisé.

Le tableau 4.2 représente le nombre des ressources occupées dans les deux architectures sur le circuit FPGA. Nous constatons que *Lifting scheme* généralisé est plus rapide et(mofadala) pour notre architecture .

	LUT	RAM	FF	IO	BUFG	Temps d'exécution (ns)
l'ondelette de HAAR par lifting schème	398 (0.75%)	20 (0.11%)	762 (0.72%)	40 (20%)	1 (3.12%)	3.722
l'ondelette de HAAR par lifting schème généralisé	453 (0,85%)	8 (0,05)	840 (0,79%)	57 (28,50%)	1 (3,13)	3.503

Tableau 4.2: Résultats de l'implémentation des deux architectures.

LUT : signifie Look Up Table, en termes généraux, est essentiellement une table qui détermine ce que la sortie est pour toute entrée (s) donnée.

RAM : pour stocker les variables utilise sans avoir à accéder à des mémoires externes

FF(flip flop) : représente les bascules.

IO: ces cellules d'entrées-sorties permettent d'interfacer le FPGA avec l'environnement extérieur.

4.5.Description du Co-design du système global

La figure 4.21 représente le Co-design du système global que nous avons réalisé :

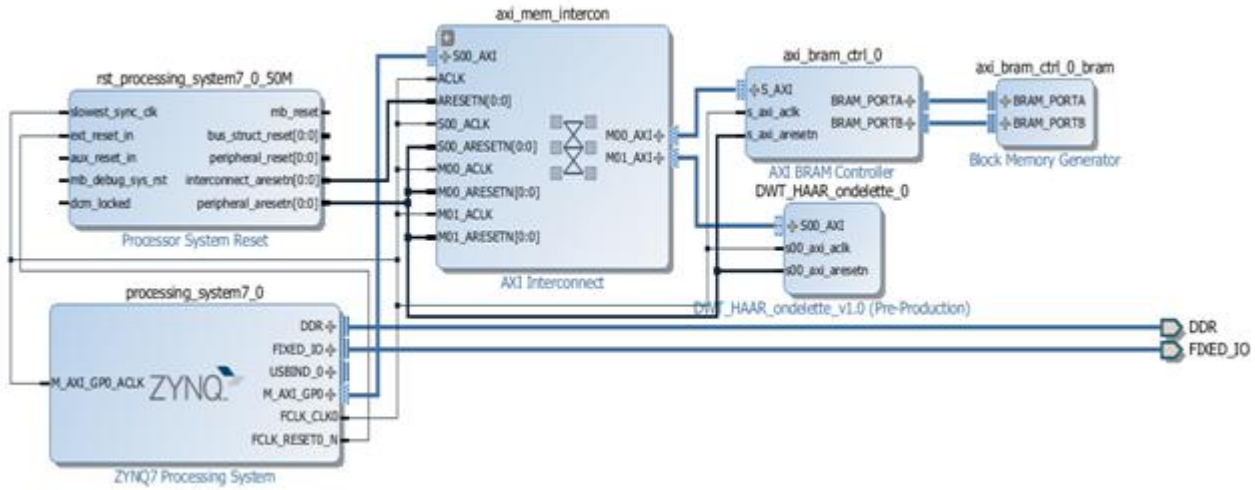

Figure 4.21: Schéma de l'architecture globale.

Les étapes pour la réalisation de notre IP sur un circuit FPGA se résume comme suit :

Configuration de la partie PS (*Programming System*). Dans notre système, cette partie possède 3 périphériques :

- Processing_systeme7_0 (Processeurs arm cortex−a9),
- Processor_system_reset,
- Axi_interconnect.

ZYNQ7 Processing System (Processeur de système ARM Cortex−A9)

Nous utilisons une configuration personnalisée du périphérique de Zynq Processing System. Les périphériques à configurer de la figure 4.22 sont :

- UART **:** qui permet la connexion en série à l'ordinateur,
- l'horloge interne : l'horloge avec laquelle le système sera cadencé,
- mémoire DDR dans laquelle est sauvegardée l'application à exécuter par le PS.

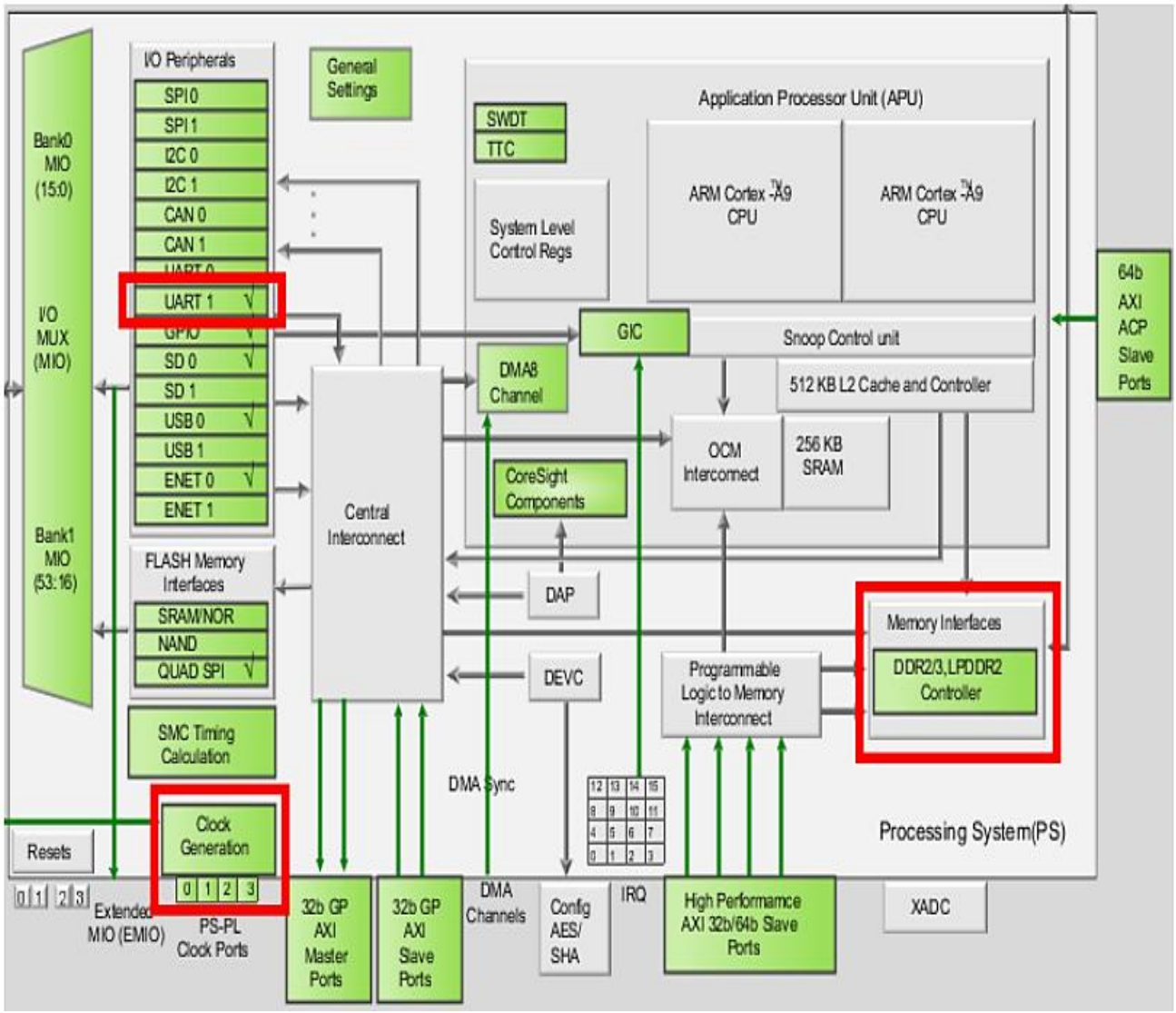

Figure 4.22: Configuration basique du périphérique PS (UART1, DDR3, clk).

- **Processor System Reset (processeur de reset)**

Il consiste à relier le reset de notre système avec le reste de la carte.

Liaison entre **PL** et **PS (AXI interface)**

On a choisi l'AXI lite dans L'AXI interface pour lier entre la partie PL et PS

4.5.1. Configuration de la partie PL (programmable Logic)

Cette partie contient un mémoire BRAM et notre IP (DWT_HAAR_ondelette_v1_0)

- **Ajout d'un périphérique PL mémoire BRAM**

Une mémoire BRAM est ajoutée à notre design à partir d'IP Catalog. Cet IP permet d'étendre l'espace mémoire total du système.

Cette mémoire peut aussi servir comme Buffer pour les données qui transitent entre PS et PL et L'AXI_BRAM_controller.

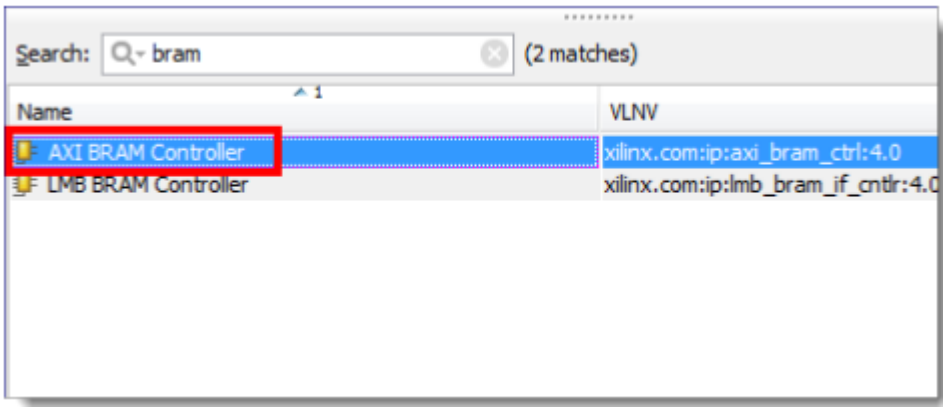

Figure 4.23: BRAM dans d'IP Catalog.

- **Ajout d'un *« custom » IP* au PL**

Le processeur ARM Cortex_9 permet de combiner un ou plusieurs IP personnalisés. Dans notre cas, un seul IP personnalisé a été ajouté. Cette partie nécessite plusieurs étapes telles que :

- Le nom défini de l'IP dans notre cas (DWT_HAAR_ondelette),
- Le nombre de registres utilisés (minimum 4),
- Aussi la taille du bus de données 32 bits.

Nous obtenons deux fichiers VHDL :

Le premier fichier (* DWT_HAAR_ondelette_v1_0.vhd) est le fichier Top Level : il contient l'instanciation de l'interface AXI, c'est le fichier dans lequel notre IP DWT_HAAR_ondelette est instancié.

Le deuxième fichier (*DWT_HAAR_ondelette_v1_0_S00_AXI.vhd) est le fichier qui définit l'interface AXI. Notre code VHDL est ajouté dans le second fichier VHD afin d'établir la connexion avec notre IP.

Le Co-design de l'IP est représenté dans la figure 2.24.

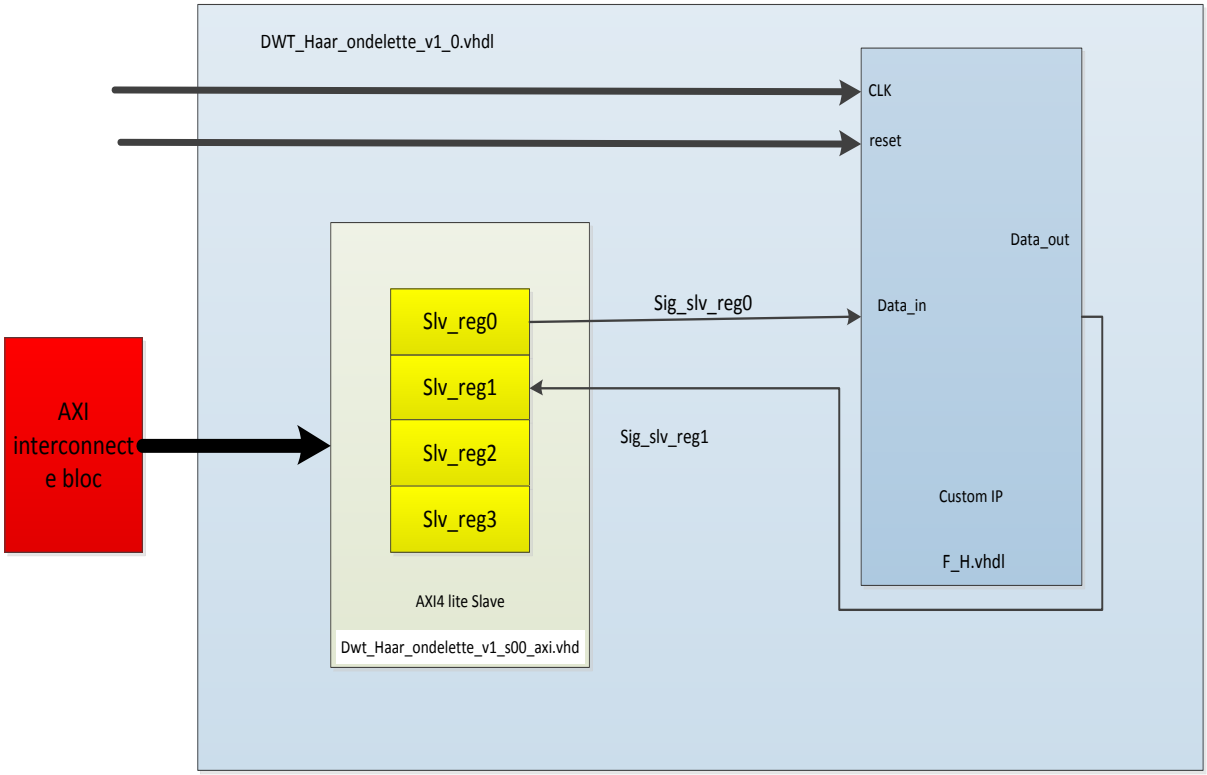

Figure 4.24: Schéma final du custom IP personnalisé.

Pour ajouter un IP à notre design, il faut tout d'abord ajouter notre IP à IP Catalog de Vivado pour nous donner la permission de l'ajouter à notre design.

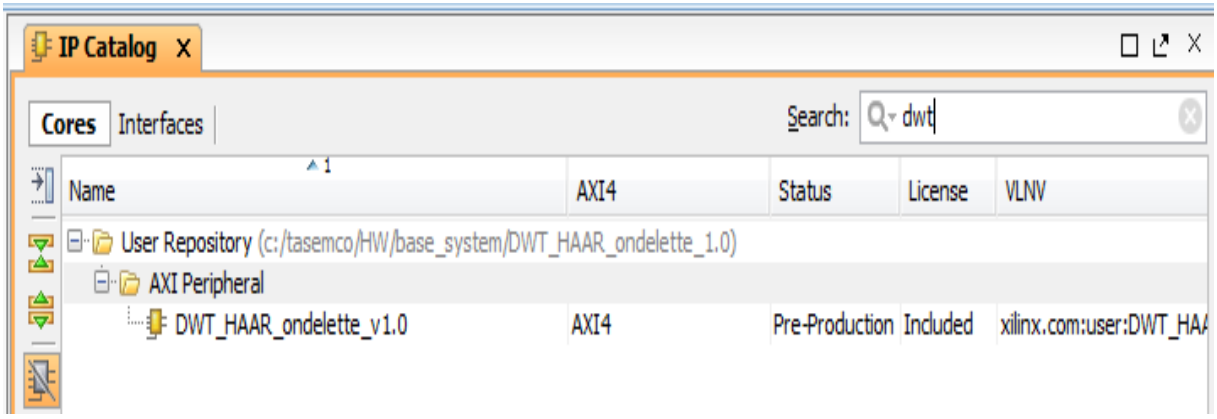

Figure 4.25: IP personnalisé et ajout à l'IP Catalog.

La plateforme hardware est configurée. Elle inclut les paramètres d'horloge et de la mémoire DDR3. Elle contient aussi un module UART et la RAM DD3 et notre IP; ensuite nous générons cette plateforme hardware et l'exportons vers SDK pour développer l'application.

L'étape HDL Wrapper génère l'interface HDL du projet, ensuite le Bitstream (il contient une logique d'interconnexions qui permet de relier les portes logiques de notre système entre elles)

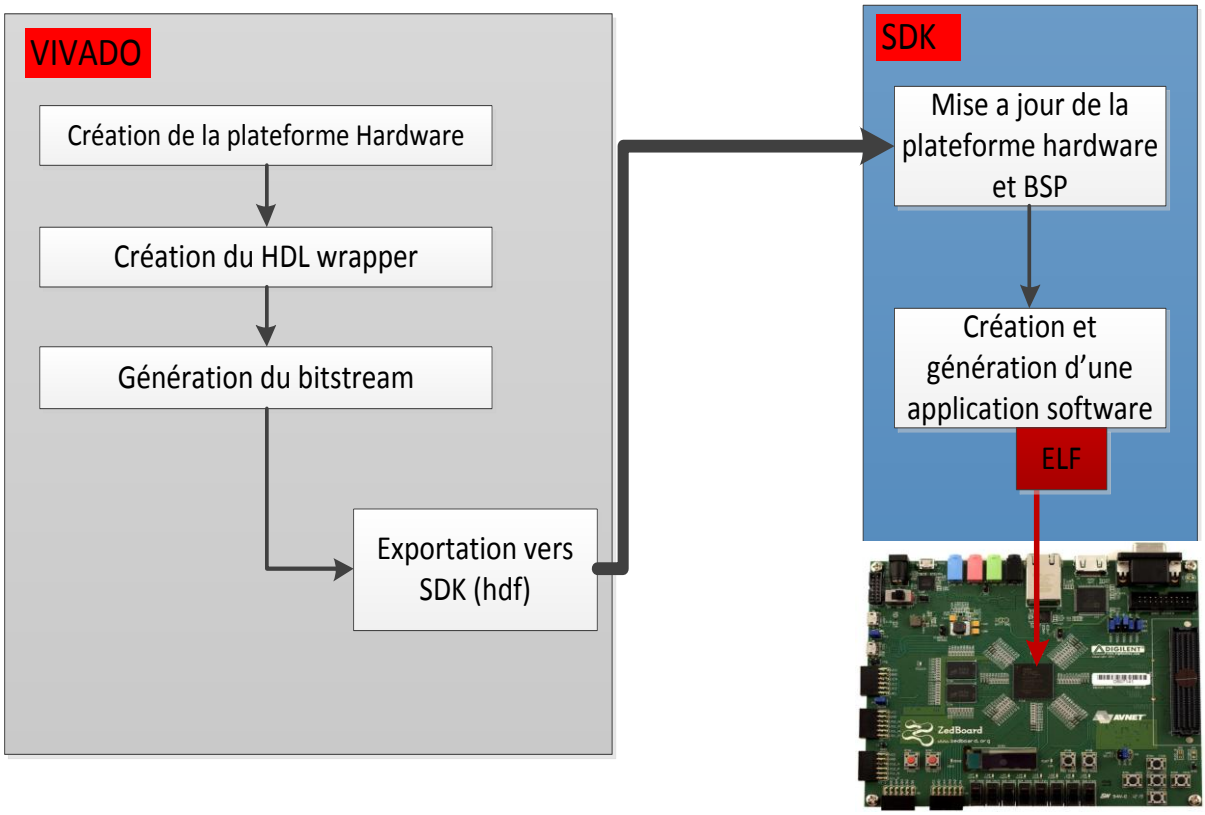

Figure 4.26: Schéma du flot de l'implémentation sur carte.

4.5.2. Génération de la plateforme Hardware et l'exportation vers SDK

Nous commençons par une plateforme matérielle Zynq. Cette plateforme matérielle définit la façon dont le PS d'ARM Cortex_9 et le PL sont configurés. Elle doit être configurée, conçue et générée avec l'outil Vivado Design Suite. Une fois achevée, les résultats peuvent être exportés de Vivado vers SDK. Un BSP (board support package) est crée afin d'intégrer les programmes C développés « *voir le code C de cette partie en annexe02* ». De la partie PS latch, le fichier ELF est transféré au niveau de la carte.

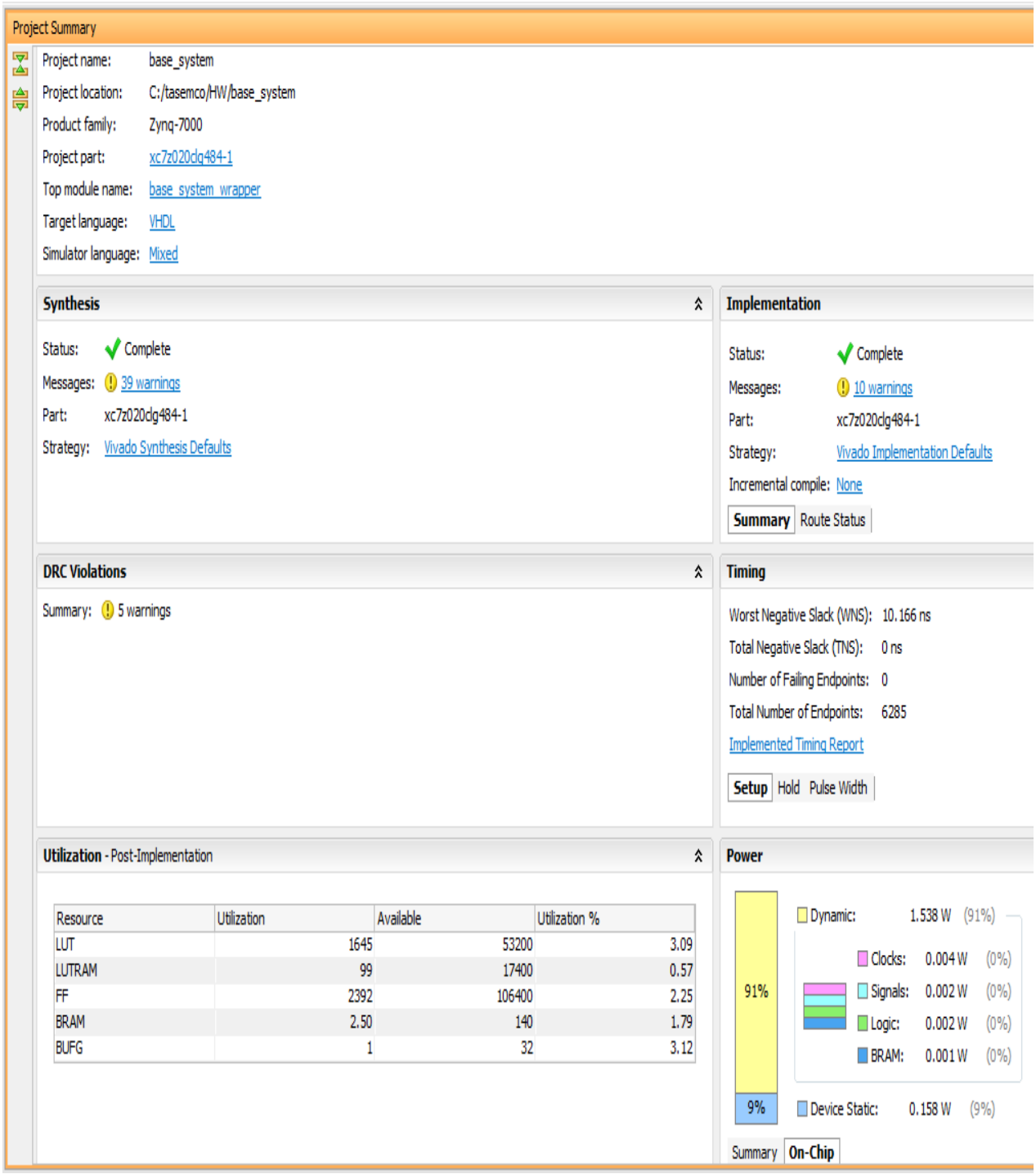

Figure 4.27: Résultat d'implémentation du Co-design.

4.6. Conclusion :

Dans ce chapitre nous avons implémenté l'approche du ***Lifting scheme*** de l'ondelette de ***HAAR*** intégrée dans la transformée de bandelette. Cette architecture a été validée sous l'outil Vivado 15.4 sur la carte ZedBoard ainsi que la réalisation d'un système embarqué de notre architecture.

Les résultats obtenus lors de la simulation montrent l'efficacité de notre approche.

Conclusion générale

De nombreuses méthodes et transformées pour l'analyse du signal existent. L'une des techniques qui a acquis une attention particulière et une reconnaissance grandissante au sein de la communauté scientifique est la transformée en ondelettes. Celle-ci est considérée comme une évolution de la transformée de Fourier.

Dans ce mémoire, nous avons étudié la transformée en bandelettes en se basant sur l'algorithme développé par ***Le Pennec*** et ***Peyré***, ensuite nous avons essayé de modifier la bandelette en remplaçant la transformée d'ondelette utilisée dans leur algorithme par une ondelette basée sur l'approche du Lifting schème, pour une meilleure implémentation matérielle sur un circuit FPGA (ZedBoard).

Pour cela nous avons défini brièvement la segmentation des images IRM et nous nous sommes intéressées aux méthodes d'ondelettes en particulier les ondelettes géométriques.

Dans le deuxième chapitre, nous avons étudié le principe de base et la théorie des bandelettes spécialement l'ondelette géométrique qui est adaptée pour une détection de contour.

Dans la troisième partie, nous avons présenté et développé sous MATLAB l'approche de *Lifting scheme* qui offre un algorithme moins complexe ce qui est le rend envisageable pour une implémentation hardware sur la carte ZedBoard.

La dernière partie a été consacrée à l'implémentation Hardware de deux architectures sur la carte ZedBoard:

- ✓ l'architecture de l'ondelette de ***Haar*** par ***Lifting Scheme,***
- ✓ l'architecture de l'ondelette par ***Lifting scheme*** généralisé.

Les résultats obtenus sont validés par plusieurs tests de simulation moyennant un environnement de simulation ISIM. Le circuit XC7Z020clg484-1 de la famille ZedBoard a été utilisé pour l'implémentation hardware : les performances ont été justifiées en termes de ressources matérielles utilisées et du temps d'exécution.

Cependant, nous estimons par ce travail avoir apporté une modeste contribution dans les domaines respectifs des transformées et de l'architecture des systèmes, et nous souhaitons que dans le futur:

- implémenter tout l'algorithme sur la partie hardware pour diminuer le temps d'exécution.
- compléter l'algorithme de bandelette par une méthode supplémentaire comme le *Level-Set* afin d'obtenir un bon résultat de segmentation

Bibliographie

Livres

[TRU 98] F.Truchete, Ondelettes pour le signal numérique, Edition Hermès, Paris 1998

[BER 12] M.Bergounioux , Mathématiques pour le traitement du signal : Cours et exercices corrigés, 2ème édition, Dunod 2012.

Cours

[MAI 00] H. Maitre, La détection des contours dans les images, 1999-2000.

[LYO 13] A.Lyonnet, M.Gardey, les ondelettes, Cours Master : Ingénierie Mathématiques Université Claude Bernard, Lyon1.

Thèses

[SOU 12] R. Soulard, Ondelettes analytiques et monogènes pour la représentation des images couleur, Thèse de Doctorat de l'université de Poitier, 2012.

[BEL 12]S.BELAROUCI et S.BENMOKHTAR, Méthode coopérative pour la segmentation d'images IRM cérébrales basée sur les techniques FCM et Level Set ,Thèse de Master de Université Abou Bekr Belkaid 2012

[MEZ 11] A.MEZIANE, Etude et comparaison des méthodes de segmentation d'images cérébrales ,Thèse de Magistère, Université Abou-Bakr Belkaid, Tlemcen,2011.

[SCH 08] B. Scherrer, Segmentation des tissus et structures sur les IRM cérébrales: agents markoviens locaux coopératifs et Formulation bayésienne. , These de Doctorat, l'INPG, 2008.

[QUE 08] Gwénolé Quellec**:** Traitement du Signal Indexation et fusion multimodale pour la recherche d'information par le contenu. Application aux bases de données d'images médicales. , Thèse de doctorat, Université de Rennes I, 2008.

[BRI 08] S.BRICQ**.** Segmentation d'images IRM anatomiques par inférence bayésienne multimodale et détection de lésions, THESE de Doctorat, Université Louis Pasteur,2008.

[SEM 07] M.Semchedine , Système Coopératif Hybride de Classification dans un SMA: Application à la segmentation d'images IRM, Thèse de Magistère, Université Farhat Abbas ,Sétif,2007.

[DOS 05] Ch.DOSSAL, Estimation de fonctions géométriques et de convolution, Thèse de Doctorat, Ecole Polytechnique, 2005.

[LAN 05] J.Landré , Analyse multi résolution pour la recherche et l'indexation d'image par le contenu dans les bases de données d'image application à la base d'images paléontologique transtyfaol , Thèse de doctorat de l'université de bourgogne en 2005.

[PEY 05] G.PEYRÉ , Géométrie multi-échelles pour les images et les textures , Thèse de Doctorat, Ecole Polytechnique en 2005.

[PEN 02] E. Le Pennec, Bandelettes et représentation géométrique des images, Thèse de Doctorat, Ecole Polytechnique, 2002.

[GER 99] L.Germond, Trois principes de coopération pour la segmentation en imagerie de résonance magnétique cérébrale, Thèse de Doctorat, L'université Joseph Fourier, 1999.

Article

[JAN 15] DJ.Jannesquin, R.dirigé, Implémentation de la Transformée en Ondelettes par l'approche *Lifting Scheme* sur GPU,2015.

[SAN 14] S. Sankar Ganesh, K. Mohanaprasad et Yepuganti Karuna ,Object Identification using Wavelet Transform , DSP Division, School of Electronics Engineering, VIT University,2014.

[NUL 11] Nul Haq K.Hayat S.H.Sherazi et W.Puech ,SEGMENTATION THROUGH DWT AND ADAPTIVE MORPHOLOGICAL CLOSING, COMSATS, Institute of Information Technology, 2011.

[LEC 08] J.Lecoeur et C.Barillot , Segmentation d'images cérébrales: Etat de l'art ,Février 2008.

[MAL 05] S.MALLAT et G.PEYRE,Orthogonal Bandlet Bases for Geometric Images Approximation, 2005.

[MAL1 02] E.Le Pennec et S. Mallat ,Geometrical Image Compression with Bandelets , 2002.

[DON 00] D.L.Donoho**,** Orthonormal ridgelet and linear singularities. Mathematical Analysis, University Press, 2000.

[CAN 00] E.Candes, D.Donoho, Curvelets: A surprisingly effective non adaptive representation of objects with edges. In Saint-Malo Proceedings.Vanderbilt, University Press, 2000.

[CAN 98] E.Candes ,Ridgelets: Theory and Applications. , Stanford University,1998.

[MAL 89] S.Mallat. A theory for multiresolution signal decomposition: the wavelet Representation. IEEE Transaction on Pattern Analysis and Machine Intelligence, 11:674–693, 1989.

Webgraphie:

[MAT 16] http://fr.mathworks.com dernière consultation le 22/06/2016.

[XIL 16] http://www.xilinx.com dernière consultation le 22/06/2016.

ANNEXE 01

- Les circuits FPGA :

Les FPGA (Field Programmable Gate Arrays ou "réseaux logiques programmables") ; littéralement traduit comme "Matrice de portes logiques programmable", sont des composants entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer certaines phases de calculs. L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents en un temps très court. Les FPGAs, nous permet de programmer du hardware à l'aide d'un langage de description matériel (VHDL). Il permet de designer et d'implémenter n'importe quelle fonction digitale.

- Le langage VHDL

C'est un langage de description de matériel qui est utilisé pour la spécification, la simulation et la preuve formelle d'équivalence de circuits. Ensuite il a été utilisé pour la synthèse automatique. C'est un langage de description de matériel pour circuit à très haute vitesse d'intégration.

Aussi le langage VHDL permet la description des aspects les plus importants d'un système matériel (*hardware system*), à savoir son comportement, sa structure et ses caractéristiques temporelles. Par système matériel, on entend un système électronique arbitrairement complexe réalisé sous la forme d'un circuit intégré ou d'un ensemble de cartes

- L'architecture d'un FPGA :

Tous les FPGA sont constitués de trois composants principaux, à savoir blocs logiques (LB), les blocs d'entrée sortie I / O, et programmable de routage ou d'interconnexion comme le montre la figure 1.1

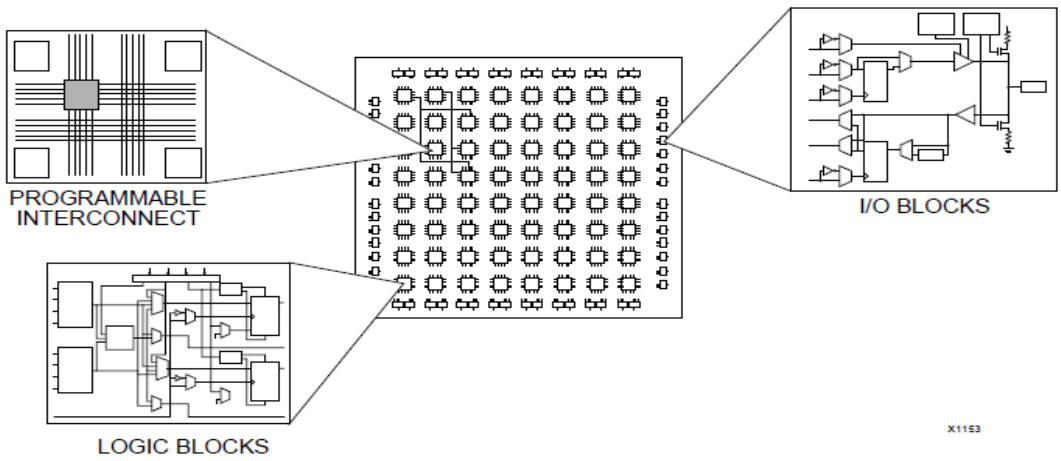

Figure 1 : Architecture générale d'un FPGA.

Les FPGA utilisent des blocs logiques complexes les CLB et des commutateurs programmables d'interconnexion. Les CLB varient d'un circuit à un autre, mais la majorité utilise des LUTs (Look Up Table).

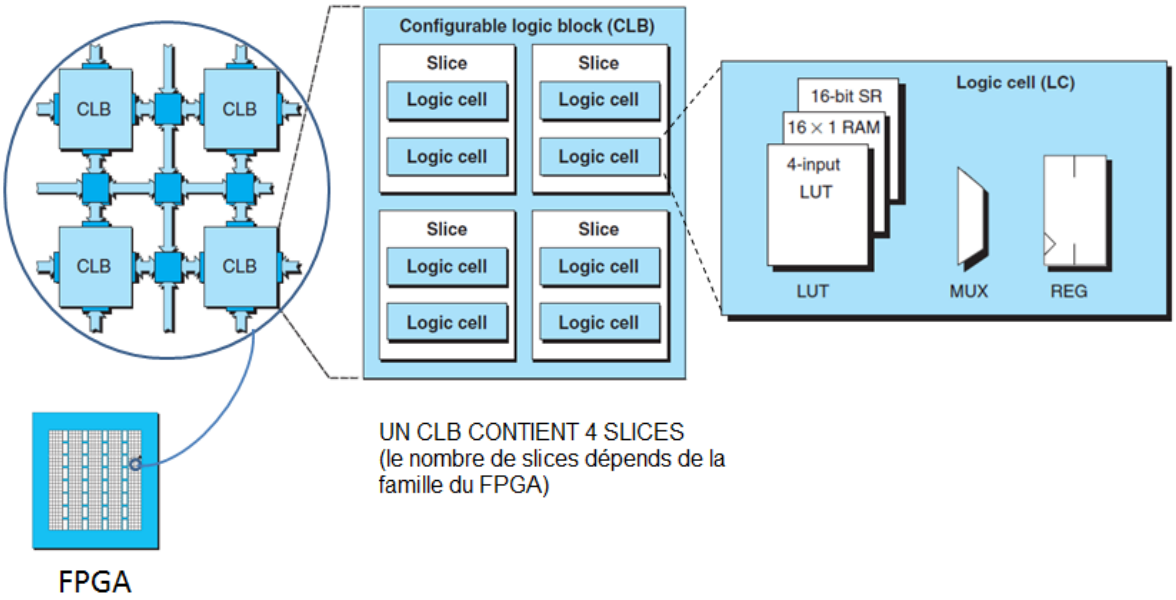

Figure 2: structure d'un CLB.

- **Les étapes du flot de conception**

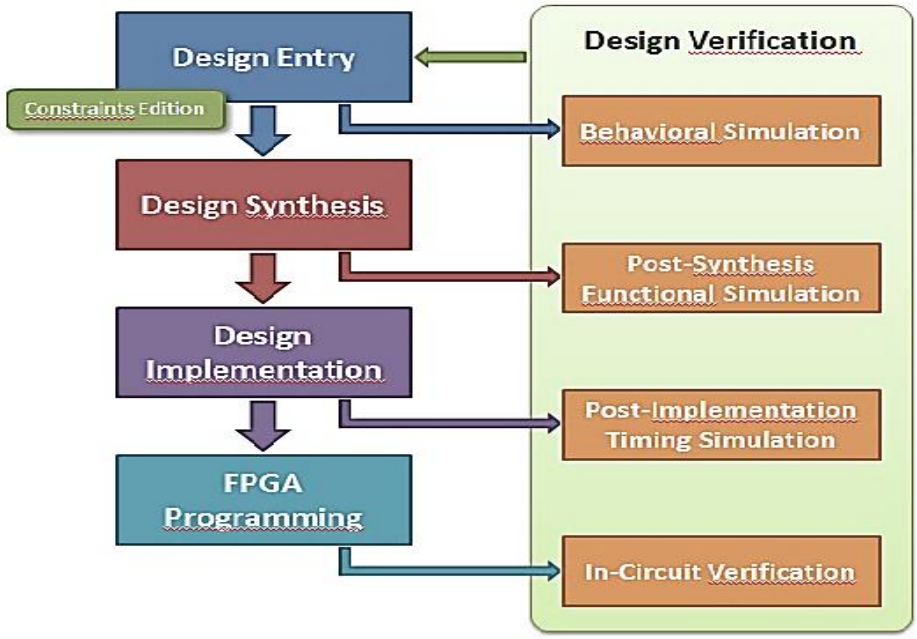

Figure 3 : Les étapes du flot de la conception.

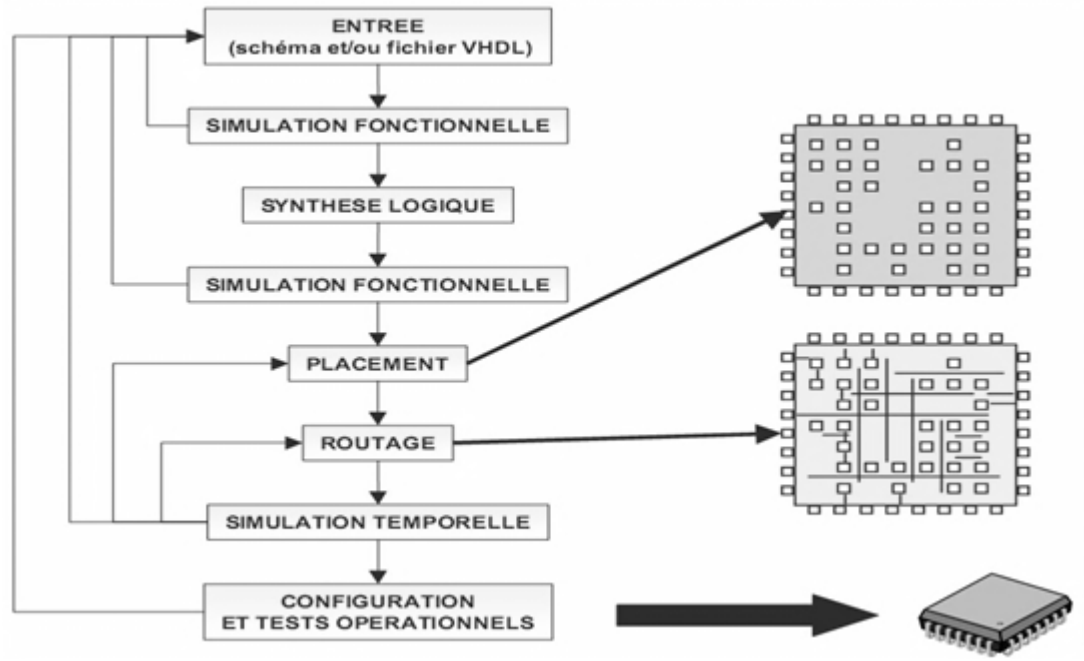

Figure 4 : Flot de conception.

➢ Evolution de la technologie FPGA

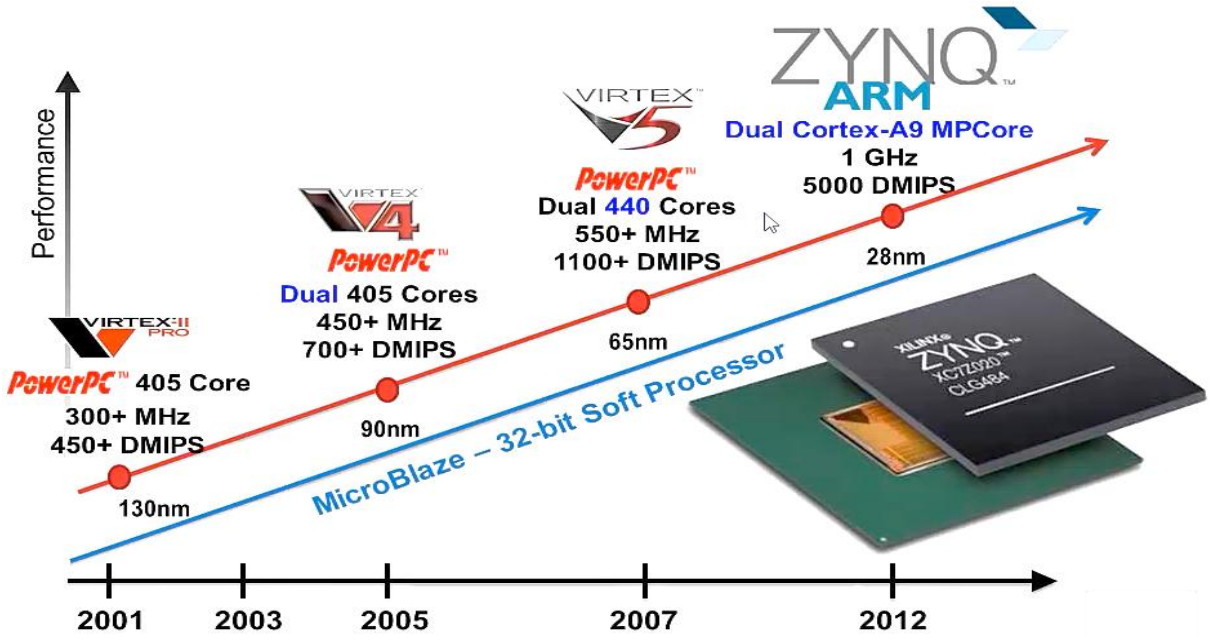

Figure 5: évolution de la technologie FPGA.

	Zynq-7000 AP SoC Devices	Z-7010	Z-7015	Z-7020	Z-7030	Z-7045	Z-7100
Processing System	Processor Core	Dual ARM® Cortex™-A9 MPCore™					
	Processor Extensions	NEON™ & Single / Double Precision Floating Point					
	Max Frequency	866 MHz			Up to 1 GHz		
	Memory	L1 Cache 32KB I / D, L2 Cache 512KB, on-chip Memory 256KB					
	External Memory Support	DDR3, DDR3L, DDR2, LPDDR2,2x QSPI, NAND, NOR					
	Peripherals	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO, 2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO					
Programmable Logic	Approximate ASIC Gates	~430K (28k LC)	~1.1M (74k LC)	~1.3M (85k LC)	~1.9M (125k LC)	~5.2M (350k LC)	~6.6M (444kLC)
	Block RAM	240KB	380KB	560KB	1,060KB	2,180KB	3,020KB
	Peak DSP Performance (Symmetric FIR)	100 GMACS	200GMACS	276 GMACS	593 GMACS	1334 GMACS	2662 GMACS
	PCI Express® (Root Complex or Endpoint)	-	Gen2 x4	-	Gen2 x4	Gen2 x8	
	Agile Mixed Signal (XADC)	2x 12bit 1Msps A/D Converter					
I/O	Processor System IO	130					
	Multi Standards 3.3V IO	100	150	200	100	212	250
	Multi Standards High Performance 1.8V IO	-	-	-	150	150	150
	Multi Gigabit Transceivers	-	4	-	4	16	16

Figure 6 : Caractéristique des différents modèles de ZYNQ.

✓ La carte ZedBoard

Figure 7 : La carte ZedBoard.

La carte ZedBoard est une carte de développement conçue par Digilent. Les systèmes on chip (SOC) ZYNQ de la société xilinx intègre :

Un système de traitement PS (processing system) basé sur un processeur double cœur ARM cortex A9 capable d'accueillir un système d'exploitation comme LINUX . On appelle PS la partie processeur et les périphériques associés cela inclus : les deux cœurs ARM cortex A9, les bus AMBA et AXI, le DMA, les GPIOs, I2C, UART, CAN, SPI, le contrôleur des mémoires QuadSPI, NAND et NOR et le contrôleur de mémoire vive.

Un système de programmation logique PL (programmable logic) avec un FPGA de la série xc7z020clg484-1. La PL, contient les éléments logiques de base, la RAM, des DSP et les entrées sorties standard.

- **Présentation des processeurs du SoC Zynq**

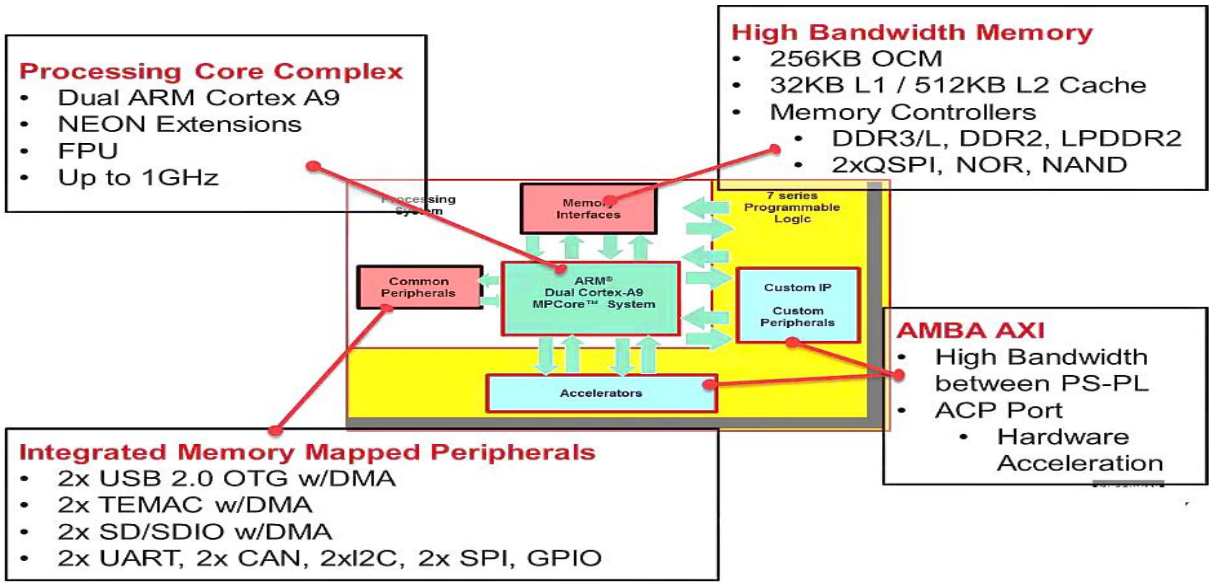

Figure 8: Présentation des processeurs du SoC Zynq.

- **Flot de conception entre la partie PS et PL**

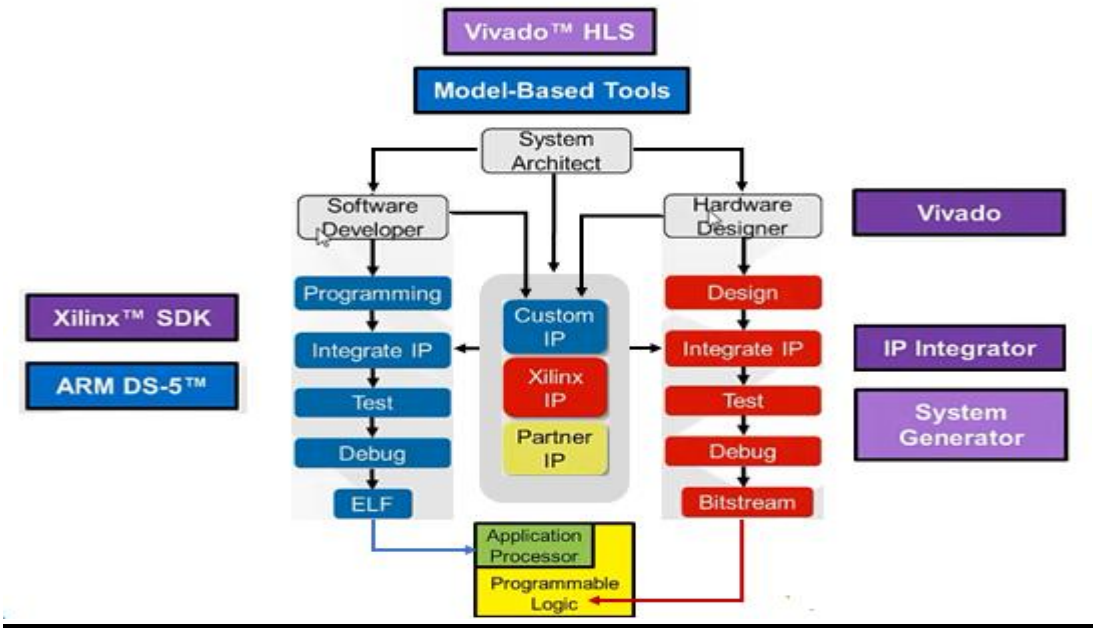

Figure 9: Flot de conception entre la partie PS et PL.

- **Les différents périphériques présents sur la ZedBoard.**

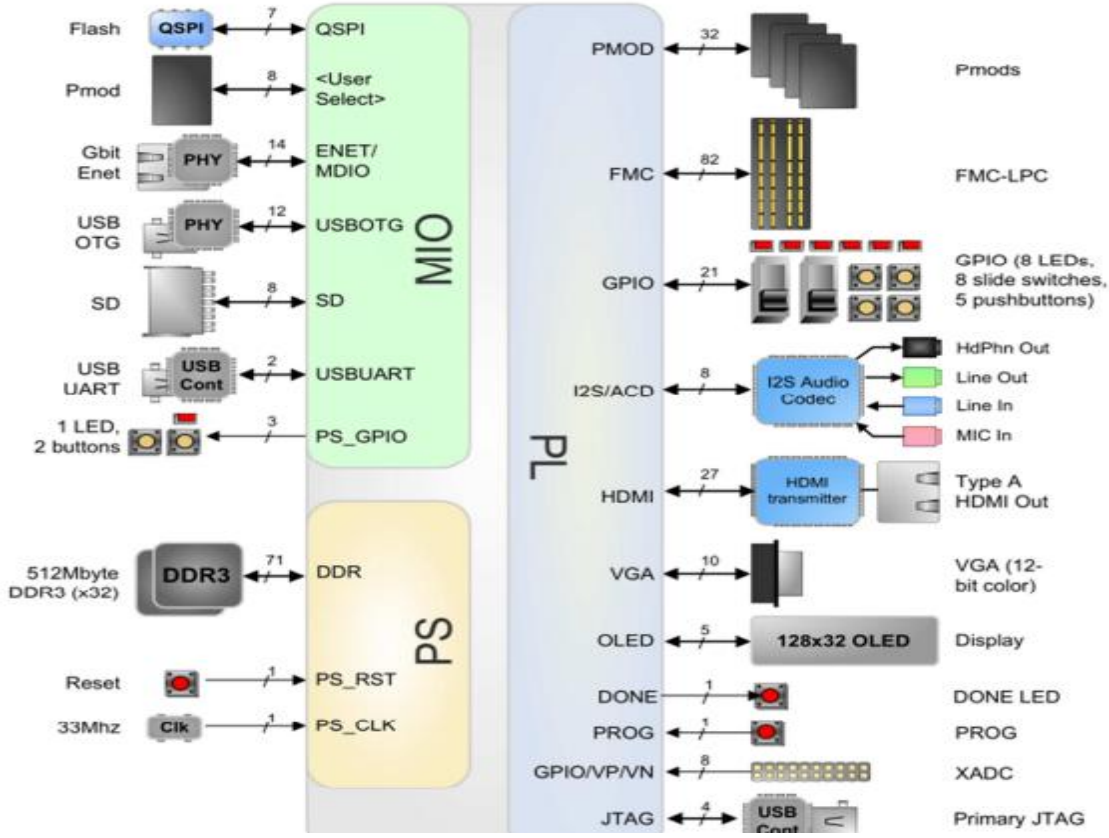

Figure 10 : Périphériques de la carte.

✓ Les outils de conception :

1- MTLAB R2009b
2- VIVADO 2015.4 et SDK
3- Xilinx ISE 13.2

ANNEXE 02

- **Le programme VHDL de l'exemple.**

```
p0_latch_pros : process(p0_latch, reset, clk, state, enable)
begin
if reset = '0' then
        p0_latch <= (others => '0');
elsif clk'event and clk = '1' then
        if state = state2 and sig_rw_ram='0' then
                p0_latch <= data_in;
        else
                p0_latch <= p0_latch;
        end if; end if; end process;
p1_latch_pros : process(p1_latch, reset, clk, state, enable)
begin
if reset = '0' then
        p1_latch <= (others => '0');
elsif clk'event and clk = '1' then
if state = state3 and sig_rw_ram='0' then
      p1_latch <= data_in;
else
        p1_latch <= p1_latch;
end if; end if; end process;
----********** div even0_1 **********
divi_even0_1 : process (clk,reset,enable)
begin
if reset ='0' then
div_even0_1 <= "000000000";
elsif rising_edge (clk) then
 if enable ='1' then div_even0_1 <= p0_latch(7)& p0_latch(7 downto 0);
end if ; end if; end process;
__***soustraction odd0_1 - div_even0_1 ****
process_soustract_odd0_1: process(clk,reset,p1_latch,div_even0_1)
begin
if reset='0' then
 sig_odd1_1<=(others=>'0');
 elsif rising_edge(clk) then
 if enable ='1' then
sig_odd1_1<= (p1_latch & '0')- div_even0_1;
 end if; end if; end process;
-- ********** div sig_odd1_1 ********
process_div_sig_odd1_1 : process (clk,reset,enable,sig_odd1_1)
begin
if reset ='0' then
div_odd1_1 <= "0000000000";
elsif rising_edge (clk) then
 if enable ='1' then
div_odd1_1<= sig_odd1_1(8) & sig_odd1_1(8 downto 0);
end if ; end if ; end process;

-- **********decalage de p0_latch**********
decalage_even0_1 : process( reset, clk, state, enable,p0_latch)
begin
        if reset = '0' then
```

```
        dec_even0_1 <= (others => '0');
        elsif clk'event and clk = '1' then
          if enable ='1' then
        if state = state5 then
        dec_even0_1 <=p0_latch &'0' & '0';
        else
        dec_even0_1 <= dec_even0_1;
        end if;end if;end if;
end process;

--        ********************* addition
sig_odd1_1 + even0_1(p0_latch) *****
addition_odd1_1_even0_1                    :
process(clk,reset,div_odd1_1,enable)
begin
if reset ='0' then
  sig_even1_1 <=(others=>'0');
   elsif rising_edge(clk) then
          if enable ='1' then
     sig_even1_1       <=       div_odd1_1+
dec_even0_1 ; end if; end if ;
end process;
```

- **Le programme VHDL d'une RAM**

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity ram is
  port (CLK ,rw_ram,reset  : in std_logic;
      addr_ram  : in std_logic_vector(6 downto 0);
      di_ram    : in std_logic_vector(11 downto 0);
      do_ram    : out std_logic_vector(11 downto 0));
end ram;
architecture synm of ram is
  type ram_type_64 is array (63 downto 0) of std_logic_vector (11 downto 0);
  signal RAM64: ram_type_64;
      signal out_ram :std_logic_vector(11 downto 0):=(others=>'0');
begin
  process (CLK ,di_ram,reset,rw_ram)
  begin
      if reset ='0' then
        out_ram <= (others=>'0');
            elsif CLK'event and CLK = '1' then
        if rw_ram = '1' then
RAM64(conv_integer(ADDR_ram)) <= di_ram;
         out_ram <= (others=>'0');
        else
            out_ram <= RAM64(conv_integer(addr_ram)) ;
            end if;
      do_ram <= out_ram;  end if ;end process; end synm;
```

- **Programme MATLAB de *Lifting Scheme* à un niveau de décomposition**

```
im=imread('Screenshot_4.png');
m=rgb2gray(I);
[Il,Ic]=size(m);FL=[];k=1;g=1;
figure;imshow(m,[]);title('Image originale');
for i=1:Il
   p1=1;p2=1;
   for j=1:Ic
      bn=i+j;
      if mod(bn,2)==0
       FL(i,p1)= m(i,j); p1=p1+1; %%les valeurs paire (even)
      else
        FL(i,(Ic/2)+p2)=m(i,j); p2=p2+1;%%les valeurs impaire(odd)
      end; end;end;
figure;imshow(FL,[]);title('Image Split sur les lignes'); FH=[0];
for i=1:Il
   p2=1;
   for j=1:Ic/2
   FH(i,(Ic/2)+p2)= FL(i,(Ic/2)+p2)-1/2*FL(i,p2); p2=p2+1;%les ligne
   end;end ;
for i=1:Il
 p1=1;
for j=1:Ic/2
 FH(i,p1)= FL(i,p1) + 1/2*FH(i,((Ic/2)+p1)); p1=p1+1;% update sur les ligne
    end;end; LL=[];
%******** split les colones ************///*****************
for j=1:Ic/2
   p1=1;p2=1;
   for i=1:Il
      bnn=i+j;
      if mod(bnn,2)==0
        LL(p1,j)=FH(i,j);p1=p1+1;
      else
        LL((Il/2)+p2,j)=FH(i,j);p2=p2+1;
      End;end;end;
for j=(Ic/2)+1:Ic
   p2=1;p1=1;
   for i=1:Il
      bm=i+j;
      if mod(bm,2)==0
        LL(p2,j)=FH(i,j);p2=p2+1;
     else
        LL((Il/2)+p1,j)=FH(i,j);p1=p1+1;
      End ;end ;end ; HH=[];

figure;imshow(LL,[]);title('Image split sur les colonnes');
%***********traitement sur les colones **********
for j=1:Ic
   p1=1;
   for i=1:Il/2
     HH(((Il/2)+p1),j)= LL(((Il/2)+p1),j)-((1/2)*LL(p1,j));p1=p1+1;
   end;end;
for j=1:Ic
   p2=1;
   for i=1:Il/2
     HH(p2,j)=LL(p2,j)+ 1/2*HH(((Il/2)+p2),j); p2=p2+1;
   end ; end
```

- **Les fonctions essentielles en C embarqué :**

```
void perform_bandelet_transform(float ** M,int ** QT,float ** Theta,int dir){
   int i,j,m=8,l=0,J_max,J_min,kx,ky, Qt[64],selx[64],sely[64];
float MB[8][8], theta,m_geom = 0;
 for(i=0;i<n;i++)
 for(j=0;j<m;j++)
 MB[i][j]=0;
   for(i=0;i<n;i++){
            for(j=0;j<m;j++){
      Qt[l]=QT[i][j];l=l+1;}}
//-------------------------------------------
J_max=max_Tableau(Qt,64);J_min=min_Tableau(Qt,64);
for(j=J_max;j>=J_min;j--){
for(kx=0,l;kx<=(n/mon_pow(2,j)-1);kx++,l++){
   for(ky=0;ky<=(n/mon_pow(2,j)-1);ky++){
      for(i=kx*mon_pow(2,j)+1,l=0;i<=(kx+1)*mon_pow(2,j);i++,l++){
         selx[l]=i;}
       for(i=ky*mon_pow(2,j)+1,l=0;i<=(ky+1)*mon_pow(2,j);i++,l++){
         sely[l]=i;}
      if (QT[kx*mon_pow(2,j)+1][ky*mon_pow(2,j)+1]==j){
      theta = Theta[kx*mon_pow(2,j)+1][ky*mon_pow(2,j)+1];m_geom = m_geom + theta;}
      else {m_geom = m_geom + gamma;}}}
return m_geom,theta;}
//-------------------------------------------------------------------------------------------------------------
void compute_best_direction(float ** M){
int i,j,ii,jj,iii,jjj,dir=1,h=9,ll=0;
float s,l,WT,t,cst,m=0,pi=3.14,sumMWT=0, theta,MW[64][64],MWt[64][64],L[64];
s=h/(2*n);t = pi/(2*n*s);
for(i=0;i<h;i++)
for(j=0;j<n;j++){
if(Theta[i][j]=theta){
   iii=0;jjj=0;
```

```
    for(ii=0;ii<h;ii++)
    for(jj=0;jj<n;jj++)     {
       cst=val_abs(-2);  cst=MW[ii][jj];
       if(val_abs(cst)<T){
            MWt[iii][jjj]=MW[ii][jj]*val_abs(cst);
           WT=mon_pow(MWt[iii][jjj],2);sumMWT=sumMWT+WT;iii=iii+1;jjj=jjj+1;}
         else if(val_abs(cst)>T){
            m=m+val_abs(cst);}}
l=sumMWT+m*mon_pow(T,2);l  = l+mon_pow(T,2);L[ll]=l;  ll=ll+1;}}
theta  = Theta(II(end));
return,theta,L;}
//*************************************************************************
void compute_wavelet_quadtree(int  Jmin),float  ** M, int Jmin,  int j_min,  int j_max){
  int  Jmax=3,j,i,ii,ll=0,q,selx[64],sely[64],  Jmin,j_min,j_max,gg;
    Jmin=1;  j_min=1;  j_max=4;
    for(j=Jmax;j>=Jmin;j--){
       if(j<Jmax)  { Jmax=j;}
     for(q=0;q<3;q++){
        if (q==0){
           for(i=1,ii=mon_pow(2,j)+1;i<=mon_pow(2,j),ii<=mon_pow(2,j+1);i++,ii++){
              selx[ll]=i;sely[ll]=ii;ll=ll+1;}}
        else if(q==1){
           for(i=1,ii=mon_pow(2,j)+1;i<=mon_pow(2,j),ii<=mon_pow(2,j+1);i++,ii++){
              selx[ll]=ii;sely[ll]=i;  ll=ll+1;}}
        else if(q==2){
for(i=mon_pow(2,j)+1,ii=mon_pow(2,j)+1;i<=mon_pow(2,j+1),ii<=mon_pow(2,j+1);i++,ii+
+){selx[ll]=ii;sely[ll]=i;ll=ll+1;gg=ll;
```

ANNEXE 02