

# Intégration d'une architecture récursive sur silicium pour la transformée en ondelettes 2D

## Silicon integration of a recursive architecture for the 2-D wavelet transform

par Camille DIOU<sup>1</sup>, Lionel TORRES<sup>2</sup>, Michel ROBERT<sup>2</sup>, Gilles SASSATELLI<sup>2</sup>

<sup>1</sup> LICM-CLOES, Université de Metz, 7 rue Marconi, 57070 Metz, France diou@sciences.univ-metz.fr tél :(+33) 3.87.54.73.06 Fax : (+33) 3. 87.54.73.01

<sup>2</sup> LIRMM-CNRS, Université Montpellier II, 161 rue Ada, 34392 Montpellier Cedex 5, France torres,robert,sassatelli@lirmm.fr tél :(+33) 4.67.41.85.75

Fax : (+33) 4. 67.41.85.00

### *résumé et mots clés*

Ce papier introduit brièvement le formalisme mathématique de la transformée en ondelettes. Nous abordons ensuite les notions d'ondelettes orthogonales et bi-orthogonales, ainsi que le traitement par bancs de filtres. Nous présentons plus particulièrement la technique du *lifting scheme* que nous comparons ensuite avec la technique des bancs de filtres, et soulignons les avantages en termes d'implantation matérielle qu'apporte le *lifting scheme* par rapport à des architectures classiques reposant sur les bancs de filtres. Nous rappelons ensuite les différents types d'architectures proposées dans la littérature pour la transformée en ondelettes, puis présentons une architecture efficace pour la transformée en ondelettes 2D exploitant la technique du *lifting scheme*. L'extension de l'architecture proposée pour la décomposition multirésolution est ensuite décrite, puis nous concluons sur les résultats obtenus et les perspectives de ce travail. L'architecture proposée autorise le traitement temps-réel des images, tout en nécessitant une surface silicium minime.

Transformée en ondelette, lifting scheme, cœur d'ondelettes.

### *abstract and key words*

This paper introduces the mathematical formalism of the wavelet transform. We present the orthogonal and bi-orthogonal wavelets, and the filter banks based signal processing. We point out the lifting scheme technique and compare it to the filter banks method, and show the advantages in terms of hardware implementation offered by the lifting scheme, compared to classical filter banks based architectures. We then review the different types of architectures for the wavelet transform found in the literature, and present an efficient architecture for the 2D wavelet transform using lifting scheme techniques. The extension of this architecture for the multiresolution decomposition is then described. The proposed architecture allows the real-time processing of images, and only needs a minimal silicon area. This architecture is suitable for embedded image processing system on chip, for video coding and compression.

Wavelet transform, lifting scheme, wavelet core.

## 1. introduction

Ces dix dernières années, une technique d'analyse du signal a acquis une reconnaissance grandissante dans la communauté scientifique internationale, de par ses qualités intrinsèques : la transformée en ondelettes, que l'on peut considérer comme une évolution de la transformée de Fourier, présente des caractéristiques intéressantes dans de nombreux domaines du traitement du signal et de l'image. Associée à l'analyse multirésolution, elle permet l'observation des images d'une manière jusqu'alors très peu employée : l'analyse ne se limite plus à l'image telle qu'elle nous apparaît, mais permet l'étude des objets présents dans l'image à différentes échelles.

Les qualités de la transformée en ondelettes lui ont valu une attention particulière de la part des scientifiques de tous les domaines : mathématiques, informatique, physique, géologie, micro-électronique, et d'une manière générale, toutes les sciences qui doivent faire appel à l'analyse de signaux. La transformée en ondelettes est donc au carrefour des sciences et trouve ses applications dans de nombreux domaines. Selon le domaine, on voudra effectuer la transformée en ondelettes de différentes manières : l'analyse statistique par exemple reposera sur les mathématiques ; les applications informatiques pour le traitement d'image (compression en vue de la transmission sur un réseau, par exemple) utiliseront bien évidemment une représentation de la transformée en ondelettes adaptée au traitement numérique. Enfin si l'on veut encore plus de performance, ou permettre la transformée en ondelettes sur des systèmes embarqués, on préférera alors une implantation totalement matérielle voire mixte (matérielle/logicielle).

Ces raisons font que nous orientons notre architecture de telle manière qu'elle puisse être réemployée dans un système de traitement d'images complet ; ainsi, dans un circuit intégré pour la compression d'image par exemple, on pourra insérer notre système de transformée en ondelettes entre la source et les modules de quantification/codage, ce de manière totalement transparente pour le système complet.

On voit ici que la transformée en ondelette est au cœur d'un domaine que l'on nommera AAAS (Adéquation Algorithme Architecture Silicium) qui représente bien une des problématiques du traitement du signal et des images aujourd'hui.

Nous présentons dans cet article, une nouvelle architecture permettant la transformée en ondelettes récursive en deux dimensions pour le traitement des images. Nous nous appuyons pour cela sur une nouvelle technique de transformée en ondelette, connue sous le nom de *lifting scheme*, qui présente des avantages importants en terme d'implantation. Nous verrons comment passer chaque étape de la conception grâce à une technique de conception incrémentale basée sur le prototypage (virtuel ou matériel) et la simulation-validation. Nous passerons ainsi de la représentation algorithmique de la transformée en ondelettes à une architecture fonctionnelle puis à un système sur puce. Avec

un flot de données de l'ordre de 200 Moctets par seconde, pour une surface de silicium de 2 mm<sup>2</sup> (en technologie CMOS 0,25 μm) l'architecture proposée correspond typiquement aux performances des supports multimédia et télécommunications de troisième génération.

Dans la section 2 nous traiterons en premier lieu des aspects théoriques de la transformée en ondelettes et des différentes approches classiques existantes. La section 3 sera consacrée à une technique particulière de décomposition de la transformée en ondelettes, appelé *lifting scheme* ; nous montrerons notamment comment n'importe quelle ondelette peut être adaptée à la méthode du *lifting scheme*. Les sections 4 et 5 seront consacrées plus à la comparaison des différents types d'architectures (bancs de filtre et *lifting scheme*). Nous montrerons en terme de puissance de calcul et d'implantation l'intérêt d'une architecture basée sur la méthode du *lifting scheme*.

Enfin les sections 6 et 7 présenteront l'architecture 2D développée, la méthodologie de conception et les résultats obtenus que nous comparerons avec les principales implémentations sur silicium de la transformée en ondelettes 2D.

## 2. de Fourier aux ondelettes

L'analyse de Fourier est une des bases majeures de la physique et des mathématiques. L'universalité du concept de fréquences sur lequel elle repose en fait un outil privilégié pour le traitement du signal. Cependant, la nature même de la transformée de Fourier occulte l'information sur le temps dans les phases du signal : en pratique, on en est donc réduit à étudier un signal soit en fonction du temps, soit en fonction des fréquences qu'il contient, sans possibilité de conjuguer les deux analyses.

Afin de pallier ce manque d'information sur le temps, une nouvelle méthode d'analyse reposant sur l'emploi d'une « fenêtre glissante » est introduite : cette fenêtre, dans laquelle on considère que le signal est localement stationnaire, est utilisée comme masque sur le signal à analyser. Cependant, la fenêtre étant définie pour tout le signal, la résolution d'analyse est fixe : la transformée ne pourra être efficace à la fois sur des signaux à faibles fréquences et sur des signaux à fréquences élevées.

La transformée en ondelettes et l'analyse multi-résolution apportent une réponse à ce problème en permettant l'analyse spatio-temporelle de signaux à différents niveaux de résolution.

La transformée en ondelettes utilise des translations et des dilatations d'une fonction fixe, l'ondelette mère, pour pouvoir analyser le signal sur toute la gamme de fréquences (dilatations) et sur toute sa durée (translation) :  $\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right)$ .

Le signal est alors transformé en une fonction de deux variables  $W(a,b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{|a|}} \int f(x) \overline{\psi_{a,b}(x)} dx$ .

Dans le domaine du traitement numérique du signal, on travaille sur des signaux discrets : il convient alors d'employer la trans-

formée en ondelettes discrète, qui translate et dilate l'ondelette selon des valeurs discrètes. Ces coefficients  $a$  et  $b$  sont discrétisés de la manière suivante :  $a = a_0^m$  et  $b = n.b_0.a_0^m$ , avec  $a_0 > 1$  et  $b_0 > 0$  fixés et appartenant à  $\mathbb{N}$ , et  $m$  et  $n$  appartenant à  $\mathbb{Z}$ . Les ondelettes sont alors définies par :

$$\psi_{m,n}(x) = \frac{1}{\sqrt{a_0^m}} \psi\left(\frac{1}{a_0^m}x - nb_0\right).$$

## 2.1. analyse multirésolution et transformée en ondelettes

En tant que telle, la transformée en ondelettes permet donc l'analyse spatio-temporelle des signaux avec une précision sur la fréquence ou le temps adaptée à la nature du signal. Cependant, l'analyse multi-résolution permet de parfaire cette étude en analysant les signaux à différents niveaux de résolution. Cette étude est à rapprocher du codage en sous-bandes qui consiste à décomposer un signal en différentes bandes de fréquences : composantes hautes fréquences d'une part, et basses fréquences d'autre part, par exemple. Afin de combiner la transformée en ondelettes et l'analyse multi-résolution, une nouvelle fonction est introduite : la fonction d'échelle, que l'on peut considérer comme la fonction complémentaire de l'ondelette, nous permettra d'extraire les composantes basses fréquences du signal, alors que l'ondelette extraira les composantes hautes fréquences.

L'analyse multirésolution permet de rapprocher l'analyse en variable d'espace avec l'analyse en variable de fréquence. Elle formalise l'idée intuitive selon laquelle tout signal peut être construit par raffinements successifs, c'est-à-dire par l'ajout de détails lorsque l'on passe d'une résolution à la suivante. D'une manière plus précise, une analyse multirésolution de  $L^2(\mathbb{R})$  est définie comme une suite de sous-espaces fermés  $V_j$  de l'ensemble des fonctions d'énergie finie  $L^2(\mathbb{R})$ ,  $j \in \mathbb{Z}$ , ayant les propriétés suivantes [1] :

1.  $V_j \subset V_{j+A} \Rightarrow$  l'approximation à la résolution  $a_{j+1}$  contient toutes les informations nécessaires pour calculer le même signal à la résolution inférieure  $a_j$ .
2.  $v(x) \in V_j \Leftrightarrow v(2x) \in V_{j+1} \Rightarrow$  si  $v(x)$  appartient à  $V_j$ , la même fonction dilatée d'un facteur 2 appartient à  $V_{j+1}$ .
3.  $v(x) \in V_0 \Leftrightarrow v(x-k) \in V_0 \Rightarrow$  si  $v(x)$  appartient à  $V_j$ , la même fonction traduite d'un facteur quelconque appartient aussi à  $V_j$ .
4.  $\bigcup_{j=-\infty}^{+\infty} V_j$  est dense dans  $L^2(\mathbb{R})$  et  $\bigcap_{j=-\infty}^{+\infty} V_j = \{0\} \Rightarrow$  chaque fonction  $f(x)$  d'énergie finie ( $\in L^2$ ) peut être approximée avec une précision arbitraire par une fonction de  $V_j$ .
5. Il existe une « fonction d'échelle »  $\varphi \in V_0$ , ayant une intégrale non nulle, telle que l'ensemble  $\{\varphi(x-k) | k \in \mathbb{Z}\}$  est une base de Riesz de  $V_0 \Rightarrow$  il existe une fonction  $\varphi$  telle que  $\{\varphi(x-k)\}$  soit une base orthonormée de  $V_0$ . La fonction  $\varphi$

et toutes ses traduits d'un facteur entier forment une base orthonormée.

Nous pouvons faire quelques observations à propos de la définition précédente. Comme  $\varphi \in V_0 \subset V_1$ , il existe une suite  $(h_k) \in l^2(\mathbb{Z})$  telle que la fonction d'échelle satisfasse :

$$\varphi(x) = \sqrt{2} \sum_k h_k \varphi(2x - k) \quad (1)$$

On constate que la définition des fonctions d'échelles est similaire à celle des ondelettes. Elles sont construites par dilatation et translation d'une fonction unique, appelée parfois « père des ondelettes ». Chaque fonction de  $V_0$  peut être écrite comme une combinaison linéaire des fonctions de base  $\varphi(x) = \sqrt{2} \sum_k h_k \varphi(2x - k)$  de  $V_1$ . Cette équation fonctionnelle porte différents noms dont les plus utilisés sont « équation de dilatation » ou « équation de raffinement ».

Si on pose  $W_j$  le complément orthogonal de  $V_j$  dans  $V_{j+1}$  :

$$V_{j+1} = V_j \oplus W_j \quad (2)$$

Chaque élément de  $V_{j+1}$  peut être écrit de manière unique comme la somme d'un élément de  $W_j$  et d'un élément de  $V_j$ . L'espace  $W_j$  contient l'information de détail nécessaire pour passer d'une approximation à la résolution  $j$  à la résolution  $j+1$ . La base orthonormale d'ondelettes associée à cette analyse multirésolution est définie par

$$\psi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} (-1)^k h_{-k+1} \varphi(2x - k) \quad (3)$$

La fonction  $\psi$  est une « ondelette » si l'ensemble des fonctions  $\{\psi(x-l) | l, j \in \mathbb{Z}\}$  est une base de Riesz de  $L^2(\mathbb{R})$ . La définition de  $\psi_{j,l}$  est similaire à celle de  $\varphi_{j,l}$  de la section précédente. Ainsi, de la même manière que précédemment, comme une ondelette est également un élément de  $V_0$ , il existe une suite  $\{g_k\}$  telle que :

$$\psi(x) = \sqrt{2} \sum_k g_k \psi(2x - k) \quad (4)$$

Les coefficients  $h_k$  et  $g_k$  sont définis par les équations suivantes :

$$h_k = \frac{1}{\sqrt{2}} \int_{-\infty}^{+\infty} \varphi\left(\frac{x}{2}\right) \overline{\varphi(x-k)} dx \quad (5)$$

$$g_k = \frac{1}{\sqrt{2}} \int_{-\infty}^{+\infty} \psi\left(\frac{x}{2}\right) \overline{\varphi(x-k)} dx \quad (6)$$

Les coefficients  $h_k$  et  $g_k$  correspondent aux coefficients d'un filtre passe-bas et d'un filtre passe-haut respectivement, et  $g_k = (-1)^k h_{-n+1}$ .

## 2.2. bancs de filtres et ondelettes orthogonales

Mallat a montré que les coefficients d'ondelettes définis par la relation  $W(a, b) = \langle f, \psi_{a,b} \rangle$  peuvent être calculés à partir d'une transformée pyramidale mise en œuvre à l'aide de filtres numériques, récursifs ou non [1]. Le principe de la transformée pyramidale consiste à décomposer le signal à analyser à l'aide d'une paire de filtres conjugués en quadrature (FCQ). L'un de ces filtres fournira les coefficients d'ondelettes (ou détails), le second les coefficients d'approximation. L'approximation est elle-même à son tour décomposée par une seconde paire de filtres, l'ensemble constituant une pyramide de filtres.

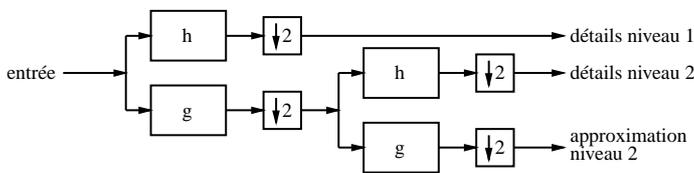


Figure 1. – Décomposition 1D à l'aide de bancs de filtres.

Soit  $s_0(n)$  un signal échantillonné correspondant au signal d'origine à la résolution 0. Ce signal est décomposé sur plusieurs niveaux de résolutions en deux bandes de fréquences (passe-haut et passe-bas) de la manière suivante, où  $s$  représente l'approximation et  $d$  les détails :

$$s_m(n) = \sum_k h(2n - k)s_{m-1}(k) \quad (7)$$

et

$$d_m(n) = \sum_k g(2n - k)s_{m-1}(k) \quad (8)$$

Ces filtres sont les filtres passe-bas et passe-haut définis par les équations 5 et 6, et sont donc liés par la même relation  $g(n) = (-1)^n h(1 - n)$ .

$s_m(n)$  est une approximation de  $s_{m-1}(n)$  à la résolution  $2^{-m}$ . Les coefficients  $d_m(n)$  représentent l'information perdue lorsque l'on passe de  $s$  à la résolution  $2^{-m+1}$  à une approximation à la résolution plus grossière  $2^{-m}$ . Les coefficients en sortie de chacun des filtres sont sous-échantillonnés avec un facteur 2 afin de respecter le théorème de Shannon. La reconstruction est obtenue de la manière suivante :

$$s_{m-1}(k) = \sum_k h(2n - k)s_m(k) + \sum_k g(2n - k)d_m(k) \quad (9)$$

## 2.3. ondelettes bi-orthogonales

La construction des FCQ doit donc répondre à un certain nombre de critères, et leur mise en œuvre est donc restreinte

pour deux raisons principales : d'une part leurs coefficients ne sont pas simples numériquement, et d'autre part ces filtres ne peuvent avoir de propriétés de symétrie, et donc ne peuvent être de phase linéaire, propriété utile en traitement d'image notamment pour éviter certains artefacts [2]. On leur préférera donc l'utilisation de filtres bi-orthogonaux explicités ci-dessous.

Dans le cas bi-orthogonal, l'analyse est effectuée de la même manière que dans le cas orthogonal [3] (cf. équations 7 et 8), mais par contre la reconstruction est assurée par les filtres conjugués, soit :

$$s_{m-1}(k) = \sum_k \tilde{h}(2n - k)s_m(k) + \sum_k \tilde{g}(2n - k)d_m(k) \quad (10)$$

La relation imposée entre les filtres étant :

$$\tilde{g}(n) = -1^n h(1 - n) \quad \text{et} \quad g(n) = -1^n \tilde{h}(1 - n) \quad (11)$$

Dans le cas bi-orthogonal, la conception des filtres est plus aisée puisqu'on permet aux filtres d'analyse de ne pas être orthogonaux entre eux.

## 3. le lifting scheme

Le *lifting scheme* a été introduit en 1994 par Wim Sweldens. Cette méthode d'implantation de la transformée en ondelettes s'affranchit totalement des notions de dilatation et de translation, et ne repose plus sur la transformée de Fourier.

### 3.1. principe

L'idée de base du *lifting scheme* est très simple [4, 5, 6]. La fonction de départ est une ondelette basique, nommée *lazy wavelet*, qui possède les caractéristiques d'une ondelette, mais dont le principal rôle consiste uniquement à séparer le signal d'entrée en deux sous-signaux ; on peut donc la considérer comme un sous-échantillonnage du signal d'entrée. Ensuite, le *lifting scheme* construit progressivement une nouvelle ondelette possédant de meilleures caractéristiques en rajoutant de nouvelles fonctions de bases. C'est ce principe d'ajout progressif des fonctions qui vaut son nom au *lifting scheme*, décrit plus précisément dans [7, 8, 9].

### 3.2. lifting scheme et matrices polyphases

La manière de procéder décrite succinctement ci-dessus n'est pas adaptée à la plupart des applications de traitement du signal ou de l'image, puisque dans la majorité des cas, on souhaiterait utiliser une ondelette déjà connue dont les propriétés sont adaptées à l'application envisagée.

Ainsi, pour utiliser n'importe quelle ondelette avec la méthode du *lifting scheme*, la solution décrite dans [7] consistera à factoriser l'ondelette en étapes élémentaires de *lifting*. Nous présentons ci-dessous un exemple concret de transformation d'un banc de filtres à l'aide des matrices polyphases ([3], et [10]).

Supposons que nous désirons utiliser les filtres suivants avec le *lifting scheme*:

$$\tilde{h}(z) = -\frac{1}{8}z^{-2} + \frac{1}{4}z^{-1} + \frac{3}{4} + \frac{1}{4}z - \frac{1}{8}z^2 \quad (12)$$

et

$$\tilde{g}(z) = -\frac{1}{2}z^{-2} + z^{-1} - \frac{1}{2} \quad (13)$$

Dans un premier temps, nous déduisons de ces filtres la matrice polyphase, donnée par l'équation 14 suivante :

$$x(z) = x_e(z^2) + z^{-1}x_o(z^2) \quad (14)$$

En l'appliquant aux deux filtres d'analyse, on obtient :

$$\tilde{h}(z) = \underbrace{\left\{ -\frac{1}{8}z^{-2} + \frac{3}{4} - \frac{1}{8}z^2 \right\}}_{\tilde{h}_e(z^2)} + \underbrace{\left\{ \frac{1}{4} + \frac{1}{4}z^2 \right\}}_{\tilde{h}_o(z^2)} z^{-1} \quad (15)$$

et

$$\tilde{g}(z) = \underbrace{\left\{ -\frac{1}{2}z^{-2} - \frac{1}{2} \right\}}_{\tilde{g}_e(z^2)} + \underbrace{\{1\}}_{\tilde{g}_o(z^2)} z^{-1} \quad (16)$$

La décomposition polyphase et l'extraction des étapes de *lifting* (dont on trouvera un développement complet dans [10]) nous donne le résultat suivant :

$$\tilde{P}(z) = \begin{pmatrix} 1 & \frac{1}{4} + \frac{1}{4}z \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -\frac{1}{2}z^{-1} - \frac{1}{2}z & 0 \\ -\frac{1}{2}z^{-1} & 1 \end{pmatrix} \quad (17)$$

L'équation 17 présente une version factorisée des filtres donnés en début de cette présentation. La figure 2 montre l'implantation de cette transformée en ondelettes dont l'algorithme de calcul est le suivant :

- séparation :  $x_k \rightarrow x_{2k}; x_{2k+1}$
- prédiction :  $d_k = -\frac{1}{2}x_{2k} + x_{2k+1} - \frac{1}{2}x_{2k+2}$
- mise à jour :  $s_k = \frac{1}{4}d_{k-1} + x_{2k} + \frac{1}{4}d_k$

De cette manière, tout banc de filtres peut être converti en étapes élémentaires du *lifting scheme*. La transformée en ondelettes étant souvent réalisée à l'aide des bancs de filtres, l'obtention des étapes élémentaires à partir des filtres se fera alors avec un minimum d'efforts.

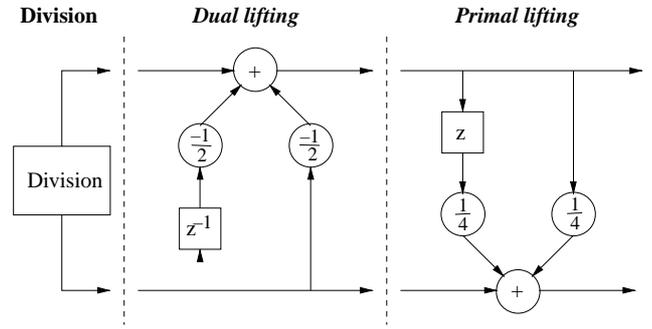


Figure 2. – Les différentes étapes du *lifting scheme* pour une prédiction linéaire.

## 4. comparaison *lifting scheme* / bancs de filtres

Si le passage d'une représentation de type « banc de filtres » à une représentation de type « *lifting scheme* » est relativement aisé, il n'est cependant pas automatique. Nous développons dans cette section les avantages intrinsèques du *lifting scheme* par rapport aux bancs de filtres, et montrons comment ceux-ci peuvent se traduire au niveau de l'architecture.

Le *lifting scheme* possède de nombreux avantages algorithmiques :

- les calculs sont effectués à la volée, ce qui permet d'économiser la mémoire nécessaire à la transformée ;
- le nombre d'opérations nécessaires au calcul de la transformée sont réduits ;
- l'algorithme présente un parallélisme SIMD ;
- la transformée inverse peut être obtenue en inversant l'ordre et le signe des opérations : la même architecture pourra donc être utilisée pour la transformée et la transformée inverse.
- le fait que la transformée soit effectuée sans aucune référence aux techniques de Fourier permet d'étendre l'application du *lifting scheme* aux signaux dont les échantillons ne sont pas placés de manière régulière, ou lorsque des contraintes telles que les effets de bord doivent être prises en compte, par exemple.

Nous détaillons ci-dessous les avantages qui se traduisent le mieux en termes d'implantation. Nous verrons ainsi pourquoi le calcul à la volée permet d'économiser de la mémoire, et comment réaliser la transformée inverse à l'aide de la même architecture.

### 4.1. calculs à la volée

Comparons la transformée en ondelettes discrète effectuée à l'aide des bancs de filtres et du *lifting scheme* pour l'ondelette

de Cohen-Daubechies-Feauveau (2,2) correspondant à la paire de filtres (5,3) [11].

Cette ondelette s'exprime de la manière suivante avec les bancs de filtres :

$$\text{Passe-haut : } d_k = -\frac{1}{2}x_{2k} + x_{2k+1} - \frac{1}{2}x_{2k+2}$$

$$\text{Passe-bas : } s_k = -\frac{1}{8}x_{2k-2} + \frac{1}{4}x_{2k-1} + \frac{3}{4}x_{2k} + \frac{1}{4}x_{2k+1} - \frac{1}{8}x_{2k+2}$$

Pour calculer  $s_k$ , nous avons besoin de connaître  $[x_{2k-2} \dots x_{2k+2}]$ . Ainsi, pour calculer  $s_{k+1}$  nous devons donc connaître  $[x_{2k} \dots x_{2k+4}]$ . La même remarque s'applique pour les  $d_k$ . Cela signifie que pour les calculs d'un échantillon, nous devons connaître tous les échantillons originaux sur la largeur du filtre : l'échantillon calculé ne peut donc pas remplacer l'échantillon original.

Dans le cas du *lifting scheme*, les calculs sont effectués de la manière suivante :

$$\text{Passe-haut : } d_k = -\frac{1}{2}x_{2k} + x_{2k+1} - \frac{1}{2}x_{2k+2}$$

$$\text{Passe-bas : } s_k = \frac{1}{4}d_{k-1} + x_{2k} + \frac{1}{4}d_k$$

Le filtre passe-haut se comporte de manière similaire à la technique des bancs de filtre, à la différence près que le signal d'origine est préalablement séparé en un sous-signal dont les échantillons sont d'indice impair ( $x_{2k+1}$ ), et un second dont les échantillons sont d'indice pair ( $x_{2k}$ ) : on a donc deux flots de données distincts, dont l'un sera utilisé pour modifier l'autre. Le calcul des  $d_k$  utilise les échantillons d'indice pair pour modifier les échantillons d'indice impair : seul un échantillon d'indice impair est nécessaire pour ce calcul, et ne sera pas réutilisé pour le calcul de l'échantillon suivant, contrairement aux bancs de filtres. On pourra donc avantageusement remplacer  $x_{2k+1}$  par la valeur calculée  $d_k$ , afin d'économiser de la mémoire.

De la même manière, le calcul des  $s_k$  nécessite un seul échantillon d'indice pair qui sera modifié à l'aide des  $d_k$  pré-calculés, et cet échantillon ne sera pas ré-utilisé pour le calcul de  $s_{k+1}$  : on pourra donc remplacer  $x_{2k}$  par  $s_k$ .

Cette caractéristique du *lifting scheme* lui confère un avantage très intéressant en terme d'implantation, avantage qui peut se traduire de 2 manières différentes :

- si les échantillons sont stockés en mémoire, on n'aura pas besoin d'une mémoire contenant les échantillons d'origine, et d'une seconde mémoire pour les échantillons transformés : tous les échantillons seront stockés dans la même mémoire ;
- les échantillons étant transformés à la volée, il devient aisé de concevoir une architecture qui effectue la décomposition en ondelettes sur le flot de données.

En d'autres termes, une architecture basée sur le *lifting scheme* est naturellement moins coûteuse qu'une architecture à base de bancs de filtres, et est plus adaptée au traitement temps-réel.

## 4.2. nombre d'opérations

Le fait de décomposer un filtre en étapes permet de réduire le nombre total d'opérations nécessaires à la transformée. Dans l'exemple précédent, l'emploi des bancs de filtres nécessite 6 additions/soustractions et 7 multiplications. La même ondelette calculée à l'aide du *lifting scheme* réduit le nombre d'opérations à 4 additions/soustractions et 4 multiplications.

Les implications sont les suivantes :

- La surface totale nécessaire à l'implantation se trouve fortement réduite, d'autant plus si le système comporte plusieurs « filtres » ;
- Le délai moyen du système est également fortement diminué, ce qui permet d'augmenter la fréquence de fonctionnement.
- La quantité de mémoire nécessaire au calcul de la transformée est également fortement réduite du fait du calcul « sur-place » des échantillons : un échantillon calculé vient remplacer l'échantillon d'origine.

## 4.3. calcul de la transformée inverse

Reprenons les différentes étapes du *lifting scheme* :

- séparation :  $x_k \rightarrow x_{2k}; x_{2k+1}$
- prédiction :  $d_k = -\frac{1}{2}x_{2k} + x_{2k+1} - \frac{1}{2}x_{2k+2}$
- mise à jour :  $s_k = \frac{1}{4}d_{k-1} + x_{2k} + \frac{1}{2}d_k$

Effectuons les étapes inverses :

- annulation de la mise à jour :  $x_{2k} = -\frac{1}{4}d_{k-1} + s_k - \frac{1}{4}d_k$
- annulation de la prédiction :  $x_{2k+1} = \frac{1}{2}x_{2k} + d_k + \frac{1}{2}x_{2k+2}$
- annulation de la séparation :  $x_{2k}; x_{2k+1} \rightarrow x_k$

Ainsi, le calcul de la transformée inverse consiste à inverser l'ordre des opérations ainsi que le signe des opérateurs. En termes d'implantation, le même système pourra donc être réemployé pour inverser la transformée en ondelettes, pour un surcoût en surface minimal (programmation des signes des opérateurs, et inversion du flot de données).

## 5. architectures pour la transformée en ondelettes 2D

Depuis les travaux de Mallat sur les bancs de filtres et l'algorithme en pyramide [1], de nombreuses architectures pour la

transformée en ondelettes sont apparues. Dans une très grande proportion, ces architectures reposent sur la technique des bancs de filtres.

La transformée en ondelettes 2D séparable consiste à effectuer deux transformées 1D selon deux directions différentes. La plupart des architectures existantes commencent par effectuer une décomposition 1D horizontalement (la transformée est effectuée selon l'ordre de lecture naturel des échantillons). L'image transformée 1D est alors mémorisée partiellement ou totalement selon le type d'architecture utilisée, puis la transformée verticale est effectuée (cf. figure 3).

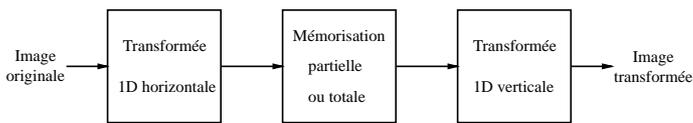


Figure 3. – Étapes de la transformée 2D.

Il existe différentes manières d'implanter la transformée en ondelettes à l'aide des bancs de filtres. On trouvera dans la littérature des architectures directes, série-parallèle, parallèle-parallèle ou à réseau systolique. Le tableau 1 résume les avantages et inconvénients de ces différentes méthodes [12, 13, 14].

Tableau 1. – Comparaison des ressources nécessaires de différents types d'architectures pour la transformée en ondelettes 2D.  $K$  représente la taille du filtre,  $J$  le nombre d'octaves, les images étant de taille  $N \times N$  [12, 13, 14].

Archi.	Mult.	Mémoire	Routage	Contrôle
Directe	$2K$	$N \times N$	Aisé	Aisé
Série-Parallèle	$4K$	$\approx 2K.N + N$	Régulier	Modéré
Parallèle	$4K$	$\approx 2K.N + N$	Régulier	Modéré
Réseau systolique	-	-	Complexe	Complexe

Si les architectures directes sont les plus aisées à mettre en œuvre, leurs performances sont directement dépendantes de la technologie employée. Ainsi, si l'architecture proposée dans [15] est caractéristique des architectures directes à structure cascadée, elle ne présente pas les performances nécessaires au traitement temps-réel des images du fait de son implantation sur des FPGA. Le circuit intégré proposé par Analog Device [16, 17], bien que reposant sur une architecture directe similaire à la précédente, est par contre plus adapté au traitement temps-réel grâce à l'emploi d'une technologie plus adéquate.

Ce manque de performances des architectures directes est dû à deux raisons principales, correspondant à deux types d'architectures :

- le premier type d'architectures directes consiste à effectuer une première transformée en ondelettes sur l'image, puis à stocker l'image transformée en mémoire avant d'effectuer les transformées des résolutions suivantes ;
- le second type d'architectures consiste à cascader autant de blocs de transformée en ondelettes que de résolutions désirées.

Ainsi, dans le premier cas, outre l'importante quantité de mémoire nécessaire, les temps d'accès à la mémoire – souvent externe – pénalisent considérablement les temps de calcul. Dans le second cas, les ressources matérielles sont multipliées et leur utilisation n'est pas optimisée.

Tous ces défauts des architectures directes font que d'autres types d'architectures leur sont souvent préférées. Beaucoup de systèmes proposés dans la littérature optimisent l'utilisation des filtres numériques afin de minimiser les ressources nécessaires à leur implantation. Dans [18], un seul filtre paramétrable est implanté, et ses coefficients sont changés alternativement entre les coefficients passe-bas et passe-haut à chaque période d'horloge, le sous-échantillonnage est alors effectué de manière implicite.

Dans [13], une méthode de partage des filtres similaire est utilisée : un même filtre peut être utilisé aussi bien pour effectuer le filtrage passe-bas que passe-haut, mais également pour effectuer les convolutions horizontales ou verticales, ce sur un ou plusieurs niveaux de résolution. Ces méthodes d'optimisation des filtres sont indépendantes du type d'architecture utilisé. Alors que dans [18] le système repose sur une architecture directe, celle employée dans [13] est à réseau systolique.

D'autres techniques que le partage de ressources peuvent être envisagées pour minimiser la taille des circuits et leur efficacité. Plutôt que de modifier l'architecture, certains auteurs proposent de modifier l'ordre de lecture des échantillons de manière à réduire la complexité du traitement et la mémoire nécessaire. Dans [19], les échantillons sont lus selon l'ordre défini par le scan de Morton. L'ordonnancement ainsi obtenu facilite grandement la décomposition 2D ainsi que la décomposition multirésolution dans le cas de l'ondelette de Haar. Cependant, l'emploi de toute autre ondelette implique une modification de l'algorithme qui complexifie grandement le contrôle du système, la taille du filtre d'ondelette n'étant alors plus forcément adaptée aux blocs définis lors de la lecture des échantillons. Un autre inconvénient de cette technique est la nécessité de stocker l'image complète en mémoire afin de pouvoir y lire les échantillons dans l'ordre désiré.

En plus des optimisations visant à minimiser les ressources nécessaires à l'implantation de la transformée en ondelettes, d'autres types d'optimisations visent naturellement à augmenter l'efficacité du traitement par bancs de filtres.

La méthode la plus évidente d'optimisation des performances consiste à maximiser la localité de l'algorithme de transformée en ondelettes. La localité étant définie par le rapport des opérations MAC sur le nombre d'accès à la mémoire externe, la solu-

tion la plus simple consiste à utiliser un ou plusieurs *buffers* pour stocker les résultats intermédiaires [20, 21]. De plus, afin de minimiser le nombre d'opérations à effectuer, il est également possible de ne pas effectuer les calculs sur les échantillons qui seront supprimés lors du sous-échantillonnage. Ceci peut être réalisé de manière implicite comme dans [18] ou commandé comme dans [20].

Tableau 2. – Caractéristiques des circuits présentés.

Circuit	Ondel.	Techno.	Surface	Fréquence Performance	Mémoire
[21]	–	0,7µm	48,4 mm <sup>2</sup>	50 MHz 1 éch./cycle	≈ 1,6 kO
[15]	SP 5 3	2 FPGA X4005		250 ns/éch. 1 éch./cycle	196 kO ext
[22]	Prog.	1,2µm	70 mm <sup>2</sup>	20 MHz 1 éch./cycle	31 O
[19]	SP 9 3	0,7µm	66,5\,mm <sup>2</sup>	1 éch./cycle	13 kO
[18]	Prog.	2 FPGA X4013	≈ 80 %	13,3 MHz 1 éch./cycle	6 × 512 kO

Cette section nous a permis de mettre en évidence les inconvénients des architectures à base de bancs de filtres. Si leur implantation reste relativement aisée, les modifications nécessaires pour en améliorer l'efficacité et/ou la surface, notamment la quantité de mémoire nécessaire, rendent leur contrôle complexe. On peut classer ces travaux selon l'approche suivie par leurs auteurs :

- *partage des ressources* : les unités arithmétiques (ou les filtres) sont exploitées au maximum de leurs possibilités, afin de ne pas gaspiller les ressources présentes dans le système ;
- *modification de l'algorithme* : l'algorithme est modifié de manière à rendre une architecture existante plus efficace ;
- *modification de l'architecture* : l'architecture est adaptée pour répondre au mieux aux besoins de l'algorithme.

Le point commun entre ces trois approches différentes est que les modifications présentées conduisent à une augmentation de la complexité du contrôle : contrôle des unités arithmétiques (modification du chemin de données), ou contrôle des données (modification des accès mémoire). Ainsi, une partie non négligeable du temps CPU est consacrée à la gestion du système plutôt qu'au calcul effectif des échantillons.

Le tableau 2 montre les surfaces et les performances de quelques circuits proposés dans la littérature. On peut constater que la surface nécessaire à leur implantation reste relativement élevée, compte tenu du fait que les mémoires nécessaires sont souvent externes.

## 6. implantation du *lifting scheme*

Dans les sections précédentes, nous avons vu que le *lifting scheme* présente des avantages importants d'un point de vue algorithmique par rapport à la technique des bancs de filtres. La plupart des avantages du *lifting scheme* se traduit directement en amélioration de l'implantation de la transformée en ondelettes.

On peut classer les architectures pour le traitement d'images en deux catégories distinctes :

- celles travaillant sur la mémoire image (*frame buffer*) : dans ce type d'architecture, les images à traiter sont préalablement stockées en mémoire, et les opérations sont effectuées sur les données présentes en mémoire ;
- celles travaillant sur le flot de données : les données sont traitées à l'issue de la source (capteur ou canal de transmission) au rythme de leur apparition ; ces systèmes doivent donc pouvoir effectuer les traitements en temps réel pour ne pas ralentir le débit des données.

Dans les deux cas de figure, le *lifting scheme* présente des avantages par rapport aux bancs de filtres. Nous avons vu dans la section 5 que le calcul des échantillons « à la volée » permet d'économiser de la mémoire : la structure du *lifting scheme* fait que les échantillons calculés remplacent les échantillons d'origine devenus inutiles ; ainsi, dans le cas du traitement sur une mémoire image, le calcul pourra donc être effectué directement sur les échantillons en mémoire. *A contrario*, l'utilisation des bancs de filtre impliquerait trois phases lors des calculs : lecture de l'échantillon, calcul de la transformée, puis écriture du résultat dans une nouvelle mémoire, l'échantillon original restant nécessaire pour le calcul de l'échantillon suivant.

Nous insisterons plus particulièrement sur le deuxième type d'architecture pour différentes raisons. Afin de faciliter la réutilisation du cœur de transformée en ondelette dans un système intégré (*design reuse*), celui-ci doit pouvoir être inséré dans une chaîne complète de traitement du signal de manière totalement transparente pour le reste de la chaîne. Ceci est facilité si la chaîne est constituée de blocs IPs travaillant sur flot de données. L'utilisation de mémoires images externes complexifie le contrôle des différentes unités ainsi que le protocole de communication entre les différents blocs, et le surcoût en surface généré par l'utilisation de mémoires internes reste un frein au développement des systèmes intégrés (SoC, *System on Chip*).

Nous présentons dans les sections suivantes la méthodologie de conception d'un cœur de transformée en ondelettes basé sur la technique du *lifting scheme*. Nous présentons rapidement l'architecture proposée pour la transformée 1D, puis développons la technique utilisée pour effectuer la décomposition 2D de manière à minimiser la mémoire nécessaire à la transformée ainsi que le contrôle des différentes unités de traitement/mémoire.

### 6.1. méthodologie de conception du cœur

Dans cette section, nous présentons la méthodologie de conception de l'architecture proposée.

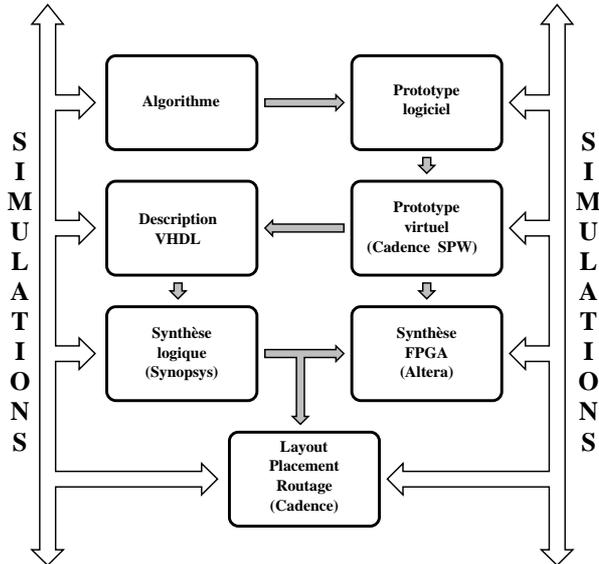


Figure 4. – Méthodologie de conception.

Dans un premier temps, nous effectuons une implantation totalement logicielle du *lifting scheme* qui ne prend en compte aucune contrainte matérielle. Le programme réalisé a pour but de valider l'architecture globale du système, en particulier le transfert des données entre les blocs de transformée 1-D horizontale et verticale, et la mémoire.

Après validation du *lifting scheme*, nous définissons plus précisément l'architecture pour la transformée en ondelettes 2D. Nous étudions notamment comment passer d'une transformée 1D à la transformée 2D en minimisant les ressources nécessaires en termes de quantité de mémoire.

Nous effectuons alors le prototypage virtuel du système à l'aide du logiciel SPW de Cadence. Cet outil nous permet de simuler le fonctionnement du système et permet sa validation fonctionnelle. Nous pouvons ensuite générer du code VHDL (comportemental ou structurel au niveau RTL) afin de passer du prototypage virtuel et de la simulation fonctionnelle à une implantation plus proche du matériel, et de permettre une caractérisation plus poussée du système.

Nous passons ensuite à la phase de synthèse logique à l'aide de Synopsys – Design Analyzer.

Le placement-routage est effectué en cellules pré-caractérisées (*standard cells*) à l'aide des outils Cadence Cell Ensemble pour la technologie CMOS 0,7 μm d'ATMEL (ECPD07, 2 niveaux de métal), et Silicon Ensemble pour la technologie CMOS 0,25 μm de ST-Microelectronics (HCMOS07, 6 niveaux de métal).

### 6.2. Architecture 1D

La figure 2 page 4 montre la simplicité d'un bloc de transformée en ondelettes. Bien évidemment, nous n'avons ici que la transformée en ondelettes 1-D sur 1 niveau de résolution. Cette implantation présente un inconvénient majeur : il comporte une avance  $z$  qui ne peut bien évidemment pas être implantée de manière matérielle. Il faut alors modifier la structure du bloc de façon à éliminer cette avance.

Reprenons les équations obtenues par *lifting scheme* pour le calcul de la prédiction et de la mise à jour :

$$\gamma_{-1,k} = \lambda_{0,2k+1} - \frac{1}{2}(\lambda_{-1,k} + \lambda_{-1,k+1}) \quad (18a)$$

$$\lambda_{-1,k} = \lambda_{-1,k} + \frac{1}{4}(\gamma_{-1,k-1} + \gamma_{-1,k}) \quad (18b)$$

On voit que l'équation 18b fait intervenir le terme  $\gamma_{-1,k-1}$  qui contient cette avance ( $k - 1$ ). Réécrivons cette deuxième équation en fonction de  $\lambda_{k+1}$  :

$$\lambda_{-1,k+1} = \lambda_{-1,k+1} + \frac{1}{4}(\gamma_{-1,k} + \gamma_{-1,k+1}) \quad (19)$$

Ceci consiste uniquement à rajouter un retard à chaque entrée du bloc de mise à jour (*cf.* figure 5).

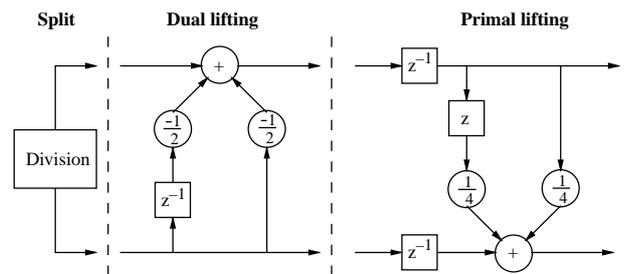


Figure 5. – Les différentes étapes du *lifting scheme* modifiées.

Cependant comme on peut le constater, le système n'est pas du tout optimisé :

- on constate dans la partie inférieure du système que les deux retards sont redondants ;
- dans la partie supérieure, il reste encore à fusionner le retard et l'avance pour faire disparaître cette dernière.

Le système final optimisé est présenté par la figure 6. On constate donc qu'avec seulement deux retards on a pu éliminer l'avance présente dans le système original. Le seul inconvénient apporté par cette modification étant l'apparition d'un délai unitaire dans le système : les échantillons en sortie sont retardés d'un coup d'horloge par rapport au système initial.

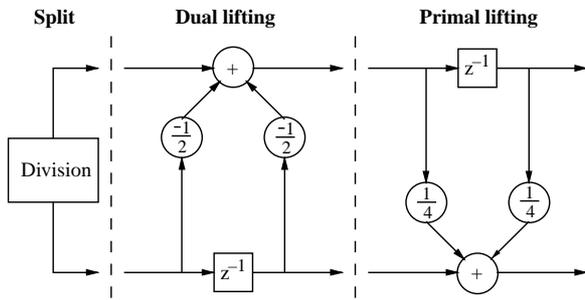


Figure 6. – Les différentes étapes du *lifting scheme* modifiées et améliorées.

### 6.3. architecture 2D

Dans le cas d'architectures travaillant sur une mémoire image, la transformée 2D ne pose pas de problème majeur : il suffit de modifier l'adressage de la mémoire pour lire les échantillons dans l'ordre désiré et effectuer la transformée en ondelettes horizontalement ou verticalement.

Dans l'architecture que nous proposons, la lecture des échantillons ne peut pas être commandée, les données devant être traitées lorsqu'elles arrivent. Les échantillons arrivant ligne par ligne, la décomposition horizontale peut-être effectuée sans problème sur le flot de données. Par contre la seconde transformée 1D étant effectuée selon l'axe des colonnes, il est nécessaire de stocker en mémoire un nombre de lignes correspondant à la taille des « filtres » utilisés.

Les différentes étapes du *lifting scheme*, présentées dans la section 3, mettent en œuvre les échantillons suivants pour effectuer la transformée :  $\gamma_k, \gamma_{k+1}, \lambda_k$  et  $\lambda_{k+1}$ . Reportés au flot de données  $s$  en entrée du système, ces 4 échantillons correspondent aux échantillons originaux  $s_{2k}, s_{2k-1}, s_{2k-2}$  et  $s_{2k-3}$ . Ainsi, pour pouvoir effectuer la décomposition verticale, il est nécessaire de mémoriser 4 lignes en sortie de la décomposition horizontale.

Nous présentons ci-dessous la méthode proposée pour effectuer la décomposition 2D de l'image sur le flot vidéo, en minimisant à la fois la quantité de mémoire nécessaire à la décomposition, ainsi que la logique de contrôle de cette mémoire.

#### 6.3.1. architecture globale du système

Après l'étude de l'algorithme permettant la décomposition 2D d'une image par *lifting scheme*, nous pouvons définir l'architecture correspondante.

L'image A est divisée en deux sous-images B et C qui après décomposition horizontale nous donnent les sous-images d'approximation (D) et de détails (E). Afin d'effectuer la décomposition verticale sur la globalité de l'image, on réunit les deux sous-images colonne par colonne (F) puis on les lit ligne par ligne pour former les sous-images G et H. Décomposées à leur

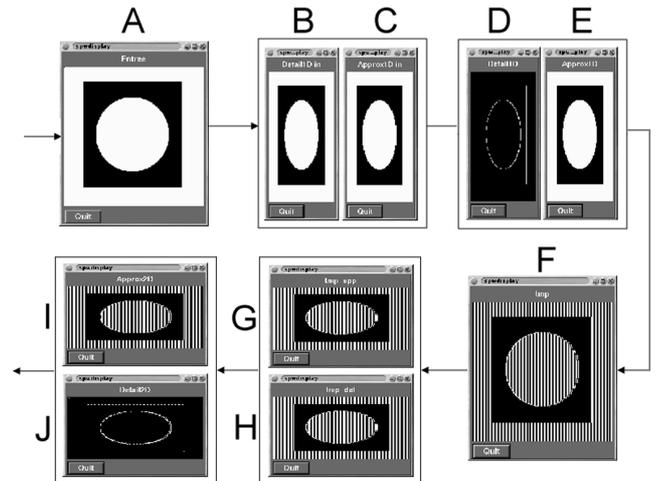


Figure 7. – Les différentes images et sous-images obtenues durant le processus de décomposition 2D.

tour verticalement ces images fournissent I et J (cf. figure 7).

La figure 8 détaille la transposition ligne/colonne permettant la décomposition verticale. Le bloc de transposition ligne/colonne est composé de 4 mémoires ligne dont la nature sera précisée par la suite. Le flot de données en entrée des mémoires  $a$  et  $b$  correspond aux lignes de l'image F de la figure 7. Ces deux mémoires sont chargées alternativement ligne par ligne puis lues alternativement pixel par pixel (donc colonne par colonne). On considère que les mémoires  $c$  et  $d$  contiennent respectivement les approximations et les détails des lignes précédentes. On calcule alors le détail du pixel courant (1) puis on met à jour le pixel précédent dans l'ordre des colonnes (2). On peut alors extraire lors de l'étape (3) les détails précédemment calculés  $d$  ainsi que l'approximation  $c$  devenus inutiles pour la suite des calculs. On recopie les approximations  $a$  et détails  $b$  vers les mémoires  $c$  et  $d$  (4), puis on incrémente le compteur colonne afin d'effectuer les opérations sur la colonne suivante (5). Lorsque toute la ligne a été traitée, les mémoires  $c$  et  $d$  contiennent les données initialement chargées dans  $a$  et  $b$  ; on recharge alors les mémoires  $a$  et  $b$  avec les nouvelles données (6).

On peut faire quelques observation sur le fonctionnement du système. Les échantillons recopiés de  $a$  et  $b$  vers  $c$  et  $d$  seront utilisés après un délai d'une ligne d'image. Les mémoires  $a$  et  $b$  servent de mémoires tampons entre le bloc de décomposition

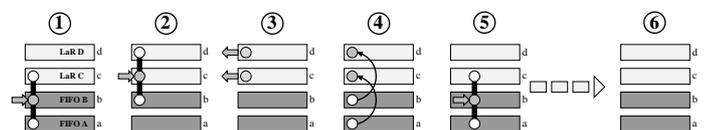


Figure 8. – Les différentes étapes de la décomposition verticale.  $a$  et  $b$  sont des mémoires tampon de type FIFOs et  $c$  et  $d$  des lignes à retard. Les ronds gris pointés par des flèches représentent le pixel dont la valeur est modifiée par les calculs.

horizontale et le bloc de décomposition verticale. La solution technologique la plus appropriée est donc d'utiliser pour les mémoires *a* et *b* des mémoires tampons de type FIFO, mais d'utiliser des lignes à retard pour les mémoires *c* et *d*. De cette manière, les échantillons copiés aux entrées de *c* et *d* seront utilisables après un délai d'une ligne, correspondant au délai entre deux pixels successifs d'une même colonne : la décomposition verticale est alors effectuée ligne par ligne également, ce qui permet le traitement des échantillons sur le flot de données à la volée.

Afin d'adapter la structure du LS-1D présentée dans la section précédente pour une décomposition selon l'axe des colonnes, il suffit donc remplacer le retard unitaire par un retard dont la profondeur correspond au nombre de pixels dans une ligne de l'image ( $k = N$ ). Ainsi, les échantillons présents en entrée d'un des additionneurs sont distants de  $N$  coups d'horloge si  $N$  est la largeur de l'image, c'est-à-dire distants d'une ligne : on a donc un échantillon et son prédécesseur selon l'axe vertical. ...tant donné que les flots de données des deux sous-images sont séparés, la distance réelle entre les deux échantillons par rapport à l'image entière est de deux lignes. Les échantillons présents sont donc bien  $x_{k-1}, x_k, x_{k+1}$  où  $k$  représente le numéro de la ligne.

En dissociant les retards nécessaires à la transformée et les blocs combinatoires, il est facile d'adapter le système complet pour différentes ondelettes et/ou tailles d'images. En modifiant la profondeur des lignes à retard, on modifie la taille des images traitées sans avoir à modifier les blocs de transformée en ondelettes, et si l'on veut modifier les fonctions de prédiction/mise à jour afin de changer d'ondelette, il n'est pas nécessaire de reconcevoir tout le système.

La figure 10 ci-dessous illustre le fonctionnement global du système.

## 7. architectures pour l'analyse multirésolution

La transformée en ondelettes présente tous ses avantages lors de l'analyse multirésolution. Dans cette section, nous présentons différentes techniques pour effectuer l'analyse multirésolution, puis présenterons plus en détails la méthode retenue.

### 7.1. techniques d'implantation pour l'analyse multirésolution

Nous avons vu que l'analyse multirésolution consiste à analyser récursivement la sortie passe-bas du système. Ainsi, afin de rendre l'architecture décrite apte à effectuer l'analyse multirésolution, plusieurs cas de figure s'offrent à nous :

- il est possible de cascader plusieurs blocs de transformée 2D. Ce principe est connu sous le nom d'algorithme en pyramide. Cette méthode ne présente aucune difficulté d'implantation, mais présente le désavantage d'être relativement coûteuse en surface, puisque le système complet est dupliqué autant de fois qu'il y a de résolutions à analyser. En appliquant cette technique à notre architecture, la taille de l'image à analyser étant divisée par 4 à chaque itération, les FIFOs et la profondeur des lignes à retard doivent donc être divisées par 2. Par contre, le fait de cascader les différents blocs produit un flot de données en sortie irrégulier, et n'exploite pas au mieux les architectures. La quantité de mémoire nécessaire à l'implantation de ce système est  $\sum_j [(\frac{3N}{2} \times \frac{1}{2^{j-1}}) + (\frac{2N}{2^{j-1}})]$  où le terme de gauche correspond à la mémoire implantée sous forme de FIFO, et le terme de droite à la mémoire implantée sous forme de ligne à retard, soit un total de  $\sum_j \frac{3,5 \times N}{2^{j-1}}$ , c'est-à-dire une quantité de mémoire inférieure à  $7N$  échantillons, quelle que soit le nombre de résolutions  $J$ .
- afin d'économiser le nombre de blocs combinatoires, il est envisageable de stocker la sous-image approximée en RAM et, dès que l'analyse de l'image originale est achevée, de poursuivre le traitement sur l'image stockée en mémoire. Il faudra bien entendu prévoir le fait que cette RAM devra être accédée

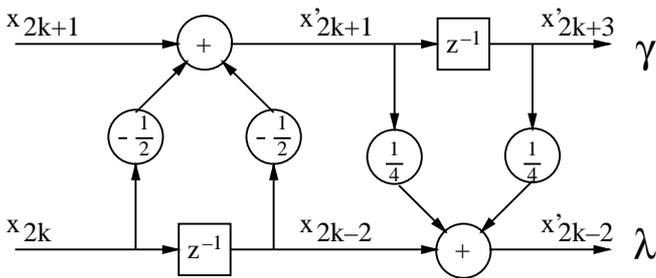


Figure 9. – Bloc de décomposition par lifting scheme.

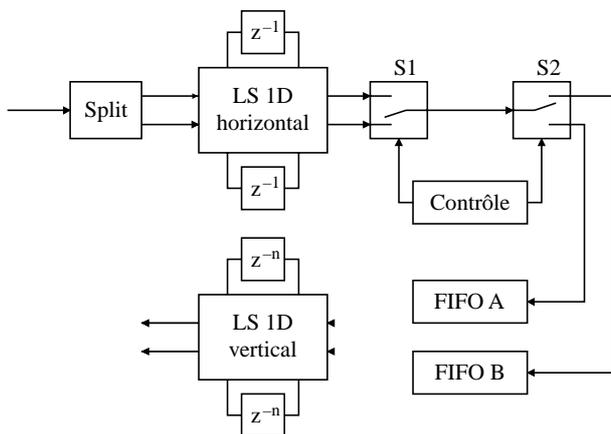


Figure 10. – Architecture pour la transformée en ondelettes 2D par lifting scheme.

en lecture par l'entrée du bloc de décomposition 2D, mais également en écriture par la sortie. L'inconvénient majeur de cette méthode est qu'elle nécessite une quantité de RAM conséquente. D'autre part, le système doit attendre que la décomposition à la résolution  $r$  soit achevée avant d'entamer la décomposition à la résolution  $r + 1$ , ce qui diminue de façon notable l'efficacité du système. La quantité de mémoire totale nécessaire est ici de  $N^2$  échantillons.

- une troisième possibilité consiste à tirer parti des caractéristiques des flots de données. En effet, les données en entrée du système sont cadencées à la fréquence  $f$ . Or le flot de données est ensuite séparé en deux : les blocs de transformée en ondelettes fonctionnent donc à la fréquence  $\frac{f}{2}$ , pour finalement fournir en sortie une image décomposée dont la fréquence pixel est de nouveau  $f$ . Il serait donc judicieux d'entrelacer les échantillons des résolutions inférieures avec ceux de l'image d'origine. Ainsi l'ensemble du système pourra fonctionner à une fréquence optimale.

## 7.2. méthode proposée

La troisième méthode de conception présentée ci-dessus présente tous les avantages nécessaires pour la conception de systèmes sur puce. Nous allons voir comment nous pouvons, en exploitant cette technique, permettre la transformée multirésolution en minimisant la quantité de mémoire nécessaire, tout en autorisant une fréquence de fonctionnement optimale.

### 7.2.1. présentation de l'algorithme en pyramide récursif

Nous avons vu précédemment que la mise en cascade de différents blocs identiques pour effectuer l'analyse multirésolution produit un flot de données en sortie irrégulier. La figure 11 illustre ce principe : si les échantillons en entrée du système sont distants de 1 à la résolution 0 (résolution initiale), la distance entre les échantillons pour chacune des résolutions suivantes est de  $2^n$ , où  $n$  est le niveau de résolution. Ainsi, le système produira d'abord un flot dont les échantillons sont distants de 2 (résolution 1), puis de 4 (résolution 2), et ainsi de suite jusqu'à

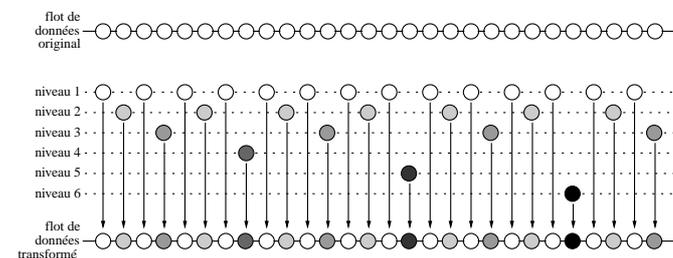


Figure 11. – Organisation des échantillons obtenue par l'algorithme en pyramide récursif : dès qu'un échantillon d'une résolution inférieure est produit, il est réinjecté dans le flot de données.

la décomposition complète du signal. La figure 11 illustre ces différents flots pour 6 niveaux de résolution. Le plus gros inconvénient de cette méthode est qu'à chaque résolution, le système concerné produit un flot de données de fréquence deux fois plus faible que le précédent, ce qui implique une sous-utilisation des unités de traitement, et donc une perte considérable de temps de calcul. Le principe de l'algorithme en pyramide récursif est donc d'effectuer les opérations sur les échantillons dès que ceux-ci sont disponibles : ainsi, dès qu'un échantillon d'une résolution inférieure est disponible, il est réinjecté en entrée du bloc de décomposition et aussitôt traité. La figure 11 illustre le flot de données produit par cette méthode : on constate que le système produit un seul flot de données dont les échantillons sont distants de 1 (à la même fréquence donc que le flot d'entrée).

### 7.2.2. présentation des architectures

Les flots de données de la figure 11 nous fournissent des indications quant à la gestion des différentes résolutions. La méthode la plus évidente consiste à utiliser un bloc de décomposition 2D entièrement consacré au premier niveau de résolution, et un second qui effectuera le traitement sur toutes les résolutions suivantes selon le principe de l'algorithme en pyramide récursif. De cette manière, les deux blocs de transformée en ondelettes 2D fonctionneront à la cadence optimale, produisant deux flots de données à la même fréquence (cf. figure 12). Le système composé des deux blocs de décomposition 2D est présenté par la figure 13.

Cependant, il est possible, si la fréquence de fonctionnement le permet, d'effectuer la décomposition multirésolution avec un seul système. En effet, la figure 14 illustre les différentes étapes de la décomposition 2D d'une image, et la fréquence des flots de données générés.

La figure 14 montre qu'à l'exception des différents *switches*, le système fonctionne à une cadence deux fois plus faible que la fréquence du signal d'entrée du système. En d'autres termes, il est donc possible en fonctionnant à cadence identique à la fréquence d'entrée d'effectuer deux fois plus d'opérations, et donc d'effectuer la multirésolution avec le même système.

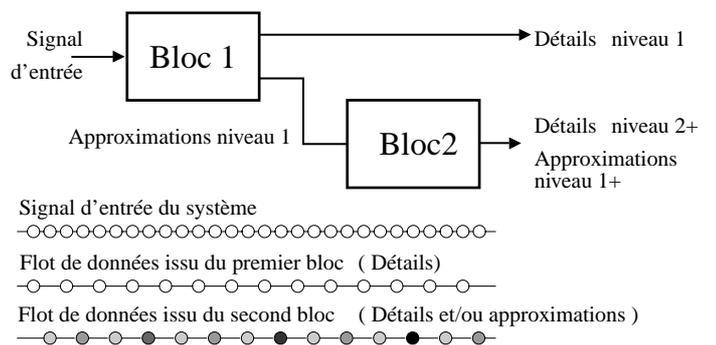


Figure 12. – Cascade de deux blocs de transformée en ondelettes 2D : le deuxième fonctionne selon le principe de l'algorithme en pyramide récursif.

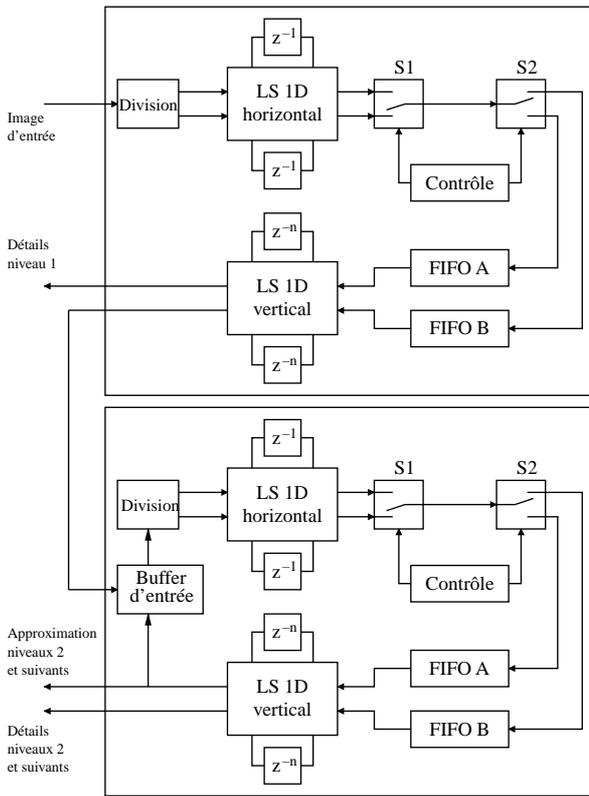


Figure 13. – Système cascadié pour la décomposition en ondelettes 2D multirésolution.

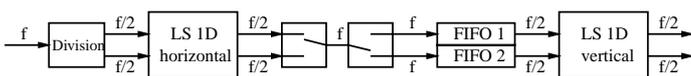


Figure 14. – Fréquences des différents flots de données dans le système pour la transformée 2D.

Afin de minimiser la quantité de mémoire nécessaire tout en conservant une fréquence de fonctionnement optimale, on prévoit plusieurs contextes de fonctionnement du système à raison d'un contexte par résolution. En terme d'implantation, tous les blocs combinatoires seront communs à tous les contextes, mais les blocs de mémorisation seront dédiés à un contexte particulier. Le principe consiste donc à dupliquer les blocs mémoires, en adaptant leurs tailles aux données à traiter. La figure 15 illustre ce principe.

La quantité de mémoire nécessaire ici est d'environ  $4Nb$  implantée sous forme de lignes à retards (décomposition verticale), et  $3Nb$  sous forme de FIFO, avec  $N =$  nombre de pixels par ligne d'image, et  $b =$  nombre de bits par pixel. Les retards nécessaires pour la décomposition horizontale sont négligés. À cette quantité de mémoire, il faudra ajouter des mémoires FIFOs utilisées comme *buffers* d'entrée du système, mémoires qu'il nous faudra caractériser. On considère dans un premier temps qu'on *bufferise* une ligne complète de chaque contexte en entrée du système : la mémoire totale nécessaire en entrée est donc de  $2Nb$ .

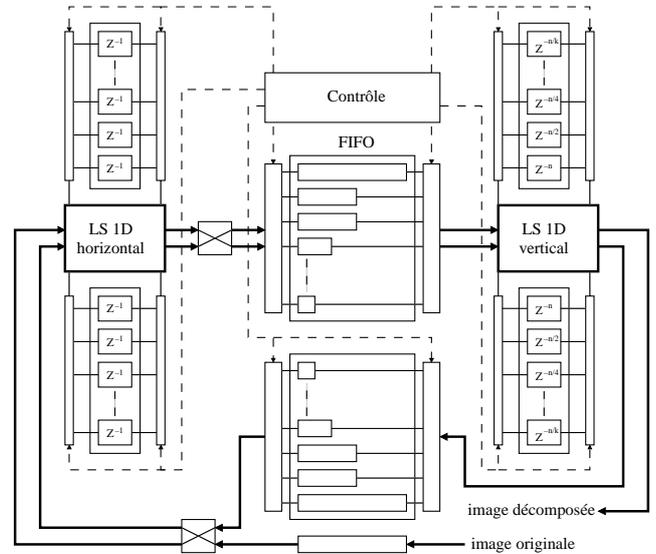


Figure 15. – Architecture monobloc pour la transformée en ondelettes multirésolution.

Ce système est donc en mesure d'effectuer la transformée en ondelettes 2D et ne nécessite qu'une quantité de mémoire limitée ( $< 9Nb$ ) qui pourra être facilement implantée sur puce afin d'économiser les temps de communication entre le système et sa mémoire, et constitue un bon compromis entre quantité de mémoire et quantité de blocs combinatoires pour une minimisation de la surface totale du système, mémoire comprise.

### 7.2.3. comparaison des deux architectures proposées

Le tableau 3 montre la différence entre les deux systèmes proposés pour la transformée en ondelettes multirésolution, comparativement au système proposé pour la transformée 2D.

La mise en cascade de deux blocs 2D double la logique nécessaire à la transformée (duplication des unités de traitement), mais permet de minimiser la taille de la mémoire tampon en entrée du second bloc, et facilite le contrôle du système, la multirésolution n'étant effectuée que par le second bloc. Cependant, on voit sur la figure 16 que la fréquence de fonctionnement des unités de traitement n'est pas optimale, puisque les échantillons sont décomposés à une fréquence deux fois plus faible que la fréquence du signal d'entrée.

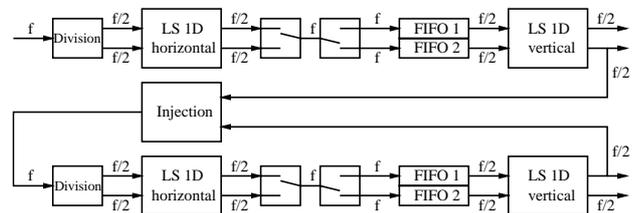


Figure 16. – Fréquences des différents flots de données dans le système pour la transformée multirésolution par mise en cascade de deux blocs.

L'utilisation d'un seul bloc pour effectuer la transformée multi-résolution implique une gestion plus lourde des données, puisque les échantillons d'approximation en sortie du système doivent être réinjectés en entrée, et insérés dans le flot de données primaire. Cependant, comme le montre la figure 17, cette structure possède l'avantage de fonctionner à une fréquence deux fois plus élevée, doublant les performances globales du système.

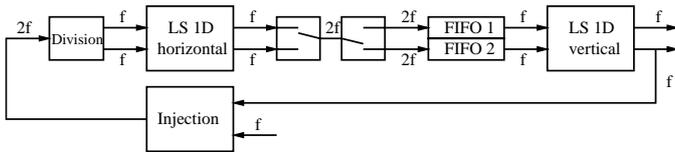


Figure 17. – Fréquences des différents flots de données dans le système pour la transformée multirésolution par système monobloc.

Tableau 3. – Comparaison des ressources nécessaires aux deux systèmes pour l'analyse multirésolution présentés par rapport au système pour la décomposition en ondelettes 2D.

Système.	Logique.	Mémoire	Tampon	Contrôle
Cascade	2 : 1	7Nb	Nb	Modéré
Monobloc	1 : 1	7Nb	2Nb	Complexe

## 8. synthèse et résultats

Nous avons présenté en préambule de ce papier le caractère multi-domaines de ce travail, au cœur même de l'AAAS (Adéquation-Algorithmes-Architecture-Silicium). L'intérêt de la transformée en ondelettes n'est plus à démontrer, et son intégration sur silicium est la cible d'un grand nombre de travaux, notamment dans le domaine des bancs de filtres.

### 8.1. une nouvelle technique de transformée en ondelettes

Ainsi, si la transformée en ondelettes par dilatation et translation de l'ondelette mère s'avère être une technique efficace et relativement aisée à mettre en œuvre par l'emploi de bancs de filtres, elle présente l'inconvénient d'être gourmande en ressources, aussi bien en mémoire qu'en surface.

Nous avons présenté les avantages que procure le *lifting scheme*, aussi bien du point de vue algorithmique que de son implantation. En effet, le principe même du *lifting scheme* permet de réduire le nombre d'opérations à effectuer pour une transformée par rapport à un traitement identique par bancs de filtres. De plus, la structure de l'algorithme permet d'effectuer les calculs sur le flot de donnée : les échantillons n'ont donc pas besoin

d'être stockés en mémoire. Enfin, contrairement à la technique des bancs de filtres qui nécessite un sous-échantillonnage après le filtrage, tous les échantillons calculés par la méthode du *lifting scheme* sont conservés. Il n'y a donc pas de perte du temps de calcul.

### 8.2. une nouvelle architecture

Les avantages résumés ci-avant ne concernent que la transformée 1-D. Or la transformée 2D nécessite beaucoup plus de ressources, et complexifie grandement les architectures pour la transformée en ondelettes. Nous avons présenté dans la section 5 différentes architectures pour la transformée en ondelettes, et nous avons vu que leur mise en œuvre n'est pas aisée.

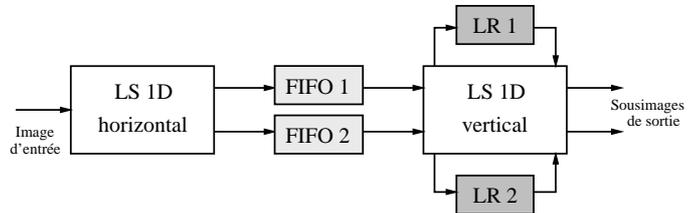


Figure 18. – Architecture proposée.

La structure du système que nous proposons et dont nous rappelons le chemin de données simplifié sur la figure 18, permet d'améliorer l'efficacité de la transformée 2D pour les raisons suivantes :

- l'algorithme du *lifting scheme* autorise le calcul des coefficients sur le flot de donnée : il n'y a donc pas de latence entre la lecture des échantillons et leur calcul effectif ;
- la structure de l'architecture globale a été conçue également pour permettre la transformée sur le flot de données : l'emploi de FIFOs comme mémoire tampon entre les blocs de décomposition horizontale et verticale permet d'effectuer la décomposition verticale parallèlement à la décomposition horizontale. Une latence de quelques lignes de l'image est alors introduite, mais peut être négligée comparativement à la taille de l'image et le nombre d'images composant la vidéo traitée ;
- le bloc de décomposition verticale utilise des lignes à retard pour synchroniser les échantillons successifs selon l'axe vertical : aucun contrôle sur ceux-ci n'est nécessaire, et le système peut donc fonctionner à flot constant.

D'autre part, la faible quantité de mémoire nécessaire à la transformée autorise son implantation directe sur la puce. La mémoire fonctionnera donc à la fréquence du système, contrairement à l'utilisation d'une mémoire externe qui impliquerait un temps d'accès très largement supérieur, et une fréquence de fonctionnement amoindrie.

Toutes ces caractéristiques font que l'architecture que nous proposons est beaucoup plus adaptée à une implantation sous forme

de cœur qu'une architecture à base de bancs de filtres. Ceci permet d'intégrer beaucoup facilement le système complet au sein même du système d'acquisition des données (capteur d'image CMOS, ou encore capteur CCD qui contrairement au précédent ne permet pas l'accès aléatoire aux pixels), ce qui a pour effet d'améliorer la rapidité du système global (capteur et traitement).

### 8.3. des performances adaptées au temps-réel

Nous rappelons le tableau 2 afin de situer le système proposé par rapport aux circuits existants dans la littérature. Bien sûr, loin d'être exhaustif, ce tableau permet de se faire une idée des ressources nécessaires à l'intégration de la transformée en ondelettes sur silicium. Nous ne rappelons ici que les systèmes intégrés sur silicium.

Tableau 4. – Récapitulatif des circuits présentés.

Circuit.	Ondel.	Techno	Surface	Fréquence Performance	Mémoire
[21]	–	0,7 $\mu\text{m}$	48,4 $\text{mm}^2$	50 Mhz	faible
[22]	Prog	1,2 $\mu\text{m}$	70 $\text{mm}^2$	20 Mhz 1 éch./cycle	faible
[19]	SP 9 3	0,7 $\mu\text{m}$	66,5 $\text{mm}^2$	1 éch./cycle	moyenne
Extrapol.	–	0,18 $\mu\text{m}$	4 $\text{mm}^2$	1 éch./cycle	moyenne

<b>Système proposé</b>	SP 5 3	<b>0,7 <math>\mu\text{m}</math></b>	<b>23,8 <math>\text{mm}^2</math></b>	1 éch./cycle	<b>faible</b>
<b>Système proposé</b>	SP 5 3	<b>0,25 <math>\mu\text{m}</math></b>	<b>2,2 <math>\text{mm}^2</math></b>	<b>200 Mhz</b> 1 éch./cycle <sup>1</sup>	<b>faible</b>
<b>Système proposé</b>	SP 5 3	<b>0,18 <math>\mu\text{m}</math></b>	<b>1 <math>\text{mm}^2</math></b> <sup>2</sup>	1 éch./cycle	<b>faible</b>

Le tableau 4 montre les gains en termes de surface ainsi qu'en quantité de mémoire du système que nous proposons pour effectuer la transformée en ondelettes 2D. Les transformées effectuées par ces différentes architectures ne sont bien évidemment pas identiques, et bien qu'elles reposent toutes sur l'utilisation de bancs de filtres, diffèrent par le type d'ondelettes employées, ou par l'aspect configurable du système.

Cependant, les valeurs données expriment bien la tendance générale de ces architectures pour la transformée en ondelettes utilisant des bancs de filtres, à savoir qu'elle reste lourde à implanter. C'est pourquoi le système que nous proposons (sur-

<sup>1</sup> Valeur minimale estimée à partir de l'analyse du chemin critique des blocs de transformée en ondelettes après optimisation.

face inférieure de moitié à technologie égale, et quantité de mémoire largement plus faible), s'avère plus intéressant dans le cadre de l'implantation de la transformée en ondelettes dans des systèmes sur puce, et de son intégration aux systèmes de prise de vue par exemple. Ces résultats sont notamment dus à l'emploi d'ondelettes adaptées à l'implantation matérielle (CDF), au type de transformée en ondelettes réalisée (*lifting scheme*), et à une technique de traitement des données à la volée originale.

La topologie du circuit dans une technologie CMOS 0,25  $\mu\text{m}$  est représentée figure 19. Une évaluation de la surface dans une technologie 0,18  $\mu\text{m}$  donne une surface de l'ordre de 1  $\text{mm}^2$ . A titre de comparaison, une extrapolation sur cette même technologie actuelle des surfaces des architectures à base de bancs de filtres nous donne une surface moyenne d'environ 4  $\text{mm}^2$ .

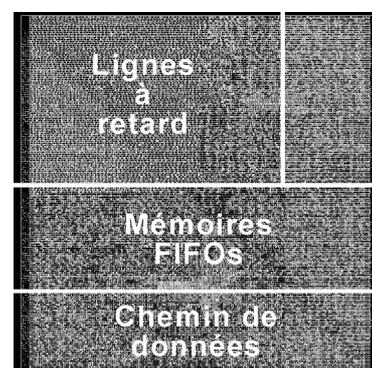


Figure 19. – Architecture synthétisée sur technologie CMOS 0,25  $\mu\text{m}$ .

## 9. conclusion et perspectives

Nous avons présenté dans cet article une technique d'implantation de la transformée en ondelettes qui présente de nombreux avantages, tant en rapidité de traitement qu'en surface silicium nécessaire.

Nous avons ainsi proposé une architecture pour la transformée en ondelettes présentant de nombreux avantages d'intégration. Ce système nous a permis de valider une nouvelle méthode de transformée en ondelettes qu'est le *lifting scheme*. Une des utilisations les plus immédiates de cette approche est sans conteste la compression d'images. Dans [23], nous avons démontré que le taux de compression que l'on pouvait obtenir à l'aide de la transformée de l'ondelette 2D permet d'avoir des facteurs de compression de l'ordre de 100 pour un rapport signal sur bruit de 30dB (sans dégradation visuelle importante de l'image). La transformée en ondelette et en particulier le *lifting scheme* ayant été choisi comme outil de transformation d'image dans la nouvelle norme de compression d'images JPEG 2000 [24], les architectures pour la transformée en ondelette devraient connaître un essor important.

<sup>2</sup> Valeur extrapolée à partir des résultats obtenus en technologie 0,25  $\mu\text{m}$ .

Le système proposé s'inscrit bien dans la tendance actuelle qui privilégie la réutilisation (*design reuse*) afin de réduire les coûts de développement. L'architecture présentée peut facilement être modifiée de manière à autoriser son intégration comme cœur de transformée en ondelette dans une architecture intégrée plus complexe (Soc : System on Chip), notamment dans le domaine du traitement d'images (capteur d'image intelligent, intégrant détection et reconnaissance de forme, par exemple). De plus, l'emploi du *lifting scheme*, de par sa nature très « modulaire », facilite également la généralisation de cette architecture à une architecture totalement reconfigurable, dynamiquement ou non. Ce type de développement des systèmes basé sur l'emploi et l'intégration de cœurs de propriétés intellectuelles (IP) se généralise dans le domaine du traitement du signal, parallèlement et de manière complémentaire à la technique privilégiant l'emploi d'architectures reconfigurables dynamiquement comme des processeurs orientés flot de données [25].

*Manuscrit reçu le 8 juin 2001*

## BIBLIOGRAPHIE

- [1] Stéphane G. Mallat, « A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 11, n°7, 1989.
- [2] A. Cohen, *Ondelettes et Traitement Numérique du Signal*, Masson, 1992, ISBN 2-225-83954-9.
- [3] Gilbert Strang and Truong Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996, ISBN 0-9614088-7-1.
- [4] W. Sweldens, « The Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Constructions », in *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. Unser, Eds. 1995, pp. 68-79, Proc. SPIE 2569.
- [5] W. Sweldens, « Wavelets and the lifting scheme: A 5 minute tour », *Z. Angew. Math. Mech.*, vol. 76 (Suppl. 2), pp. 41-44, 1996.
- [6] W. Sweldens and P. Schröder, « Building your own wavelets at home, in *Wavelets in Computer Graphics*, pp. 15-87. ACM SIGGRAPH Course notes, 1996.
- [7] I. Daubechies and W. Sweldens, Factoring Wavelet Transforms into Lifting Steps, *J. Fourier Anal. Appl.*, vol. 4, n°3, pp. 245-267, 1998.
- [8] W. Sweldens, « The lifting scheme: A custom-design construction of biorthogonal wavelets », *Appl. Comput. Harmon. Anal.*, vol. 3, n°2, pp. 186-200, 1996.
- [9] W. Sweldens, « The lifting scheme: A construction of second generation wavelets », *SIAM J. Math. Anal.*, n°2, vol. 29, pp. 511-546, 1997.
- [10] C. Valens, « The Fast Lifting Wavelet Transform », 1999, URL: <http://perso.wanadoo.fr/polyvalens/clemens/lifting/lifting.html>.
- [11] A. Cohen and I. Daubechies and J. Feauveau, « Bi-orthogonal bases of compactly supported wavelets », *Comm. Pure Appl. Math.*, vol. 45, pp. 485-560, 1992.
- [12] Chaitali Chakrabarti and Mohan Vishwanath and Robert Michael Owens, « Architectures for the Wavelet Transforms : A Survey, *Journal of VLSI Signal Processing*, vol. 14, pp. 171-192, 1996.
- [13] Jimmy C. Limqueco and Magdy A. Bayoumi, « A VLSI Architecture for Separable 2-D Discrete Wavelet Transform, *Journal of VLSI Signal Processing*, vol. 18, pp. 125-140, 1998.
- [14] Mohan Vishwanath and Robert Michael Owens and Mary Jane Irwin, « VLSI Architectures for the Discrete Wavelet Transform », *IEEE Transactions on Circuits and Systems – II : Analog and Digital Signal Processing*, vol. 42, n°5, pp. 305-316, Mai 1995.
- [15] Frédéric Truchetet and André Forsys, « Implementation of still-image compression-decompression scheme on fpga circuits, *SPIE*, vol. 2669, pp. 66-75, Janvier 1996.
- [16] *Low Cost Multiformat Video Codec ADV601*, 1997, Datasheet, URL: <http://products.analog.com>,
- [17] *Ultralow Cost Video Codec ADV601LC*, 1997, Datasheet, URL: <http://products.analog.com>.
- [18] Philippe Guermeur and Stéphane Guermeur, « Une implantation de la transformée en ondelettes discrètes à base de circuits FPGA, in *Actes des Journées Adéquation Algorithme Architecture en Traitement du Signal et Images*, Saclay, France, 28-30 janvier 1998, pp. 201-206.
- [19] Gauthier Lafruit and Francky Catthoor and Jan P. H. Cornelis and Hugo J. De Man, « An Efficient VLSI Architecture for 2-D Wavelet Image Coding with Novel Image Scan », *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, n°1, Mars 1999.
- [20] I. Urriza and J. I. Artigas and J. I. Garcia and L. A. Barrag-n and D. Navarro, « Locality Improvement of the Pyramid Algorithm to Compute the Discrete Wavelet Transform », in *Proceedings of XIII Design of Circuits and Integrated Systems Conference*, Novembre 1998, pp. 30-35.
- [21] D. Navarro, « VLSI Implementation of 2-D Mallat's Wavelet Transform.
- [22] Aleksander Grzeszczak and Mrinal K. Mandal and Sethuraman Panchanathan and Tet Yeap, « VLSI Implementation of Discrete Wavelet Transform », *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 4, n°4, Décembre 1996.
- [23] Camille Diou, *Contribution à l'intégration sur silicium de la transformée en ondelettes 2D*, Thèse de Doctorat, Université de Montpellier II, 2000.
- [24] JPEG, « JPEG 2000 Part I Final Committee Draft Version 1.0 », Tech. Rep. ISO/IEC JTC1/SC29 WG1, Avril 2000, Information Technology, JPEG 2000 Image Coding System.
- [25] Gilles Sassatelli and Lionel Torres and Pascal Benoît and Thierry Gil and Camille Diou and Gaston Cambon and Jérôme Galy, « Highly Scalable Dynamically Reconfigurable Systolic Ring-Architecture for DSP Applications, in *Proceedings DATE'02*, IEEE Design, Automation and Test in Europe, 4-8 mars 2002.

## LES AUTEURS

### Michel ROBERT



Michel ROBERT, 44 ans, est Professeur à l'université de Montpellier 2. Il enseigne la CAO des circuits et des systèmes intégrés micro-électroniques à l'ISIM (Institut des Sciences de l'Ingénieur de Montpellier). Après une expérience industrielle d'ingénieur d'étude à Texas Instruments, il débute sa carrière d'enseignant-chercheur à l'université Montpellier 2 en octobre 1984 (thèse de doctorat soutenue en 1987). Il effectue ses travaux de recherche au LIRMM (laboratoire d'informatique, de robotique et de micro-électronique) dans le domaine de la conception et de la validation des systèmes intégrés micro-électroniques, la synthèse électrique et topologique des circuits intégrés numériques CMOS, et l'adéquation algorithme-architecture en traitement d'images temps réel. Il est auteur ou co-auteur de plus de 150 publications dans ces domaines. Nommé membre de l'Institut Universitaire de France en 1997, il est Vice-président du CNFM (Comité National de Formation en Micro-électronique) depuis 1999, conseiller au ministère de la Recherche (membre du Conseil Supérieur de la Recherche et de la Technologie) depuis 1999, chargé de mission au département STIC du CNRS depuis 2001 et directeur de l'école doctorale I2S (Information Structures et Systèmes) depuis 2001.

### Lionel TORRES



Lionel TORRES a obtenu son doctorat en Micro-électronique en 1996 à l'université Montpellier2. De 1996 à 1997, il a travaillé au sein de la société ATMEL en tant qu'ingénieur de conception. Depuis 1998 il est maître de conférences à l'université Montpellier 2 et il effectue sa recherche au Laboratoire d'informatique, de Robotique et Micro-électronique de Montpellier. Ses recherches au sein du département micro-électronique concernent les architectures micro-électroniques pour le traitement du signal et des images, et les architectures reconfigurables dynamiquement.

### Camille DIOU



Camille DIOU a obtenu son doctorat en Micro-électronique en 2000 à l'université Montpellier 2. Depuis 2001, il est Maître de conférences à l'université de Metz où il enseigne la CAO micro-électronique et les systèmes numériques. Les travaux de recherche qu'il effectue au Laboratoire Interfaces Capteurs et Micro-électrique de Metz (LICM) dans le domaine des systèmes électroniques pour les télécommunications, portent sur la conception d'architectures fortement parallèles de circuits et systèmes pour la transmission d'information à haut débit et la modulation numérique.

### Gilles SASSATELLI



Gilles SASSATELLI a obtenu son doctorat en Micro-électronique en 2002 à l'université Montpellier 2, au sein du Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier (LIRMM). Il est actuellement professeur assistant au laboratoire MES (MicroElectronics Systems) de l'université de Darmstadt en Allemagne où il poursuit des travaux sur les architectures reconfigurables dynamiquement dédiées au traitement du signal et des images.