

# **Evaluation des Transformées Orthogonales par l'Arithmétique Half-line en $O(\log L)$**

par

**Hamid BESSALAH**

Centre de Développement des Technologies Avancées  
128, chemin Mohamed Gacem, BP-245,

El Madania, Alger, Algérie

Tél: (213)-2-65-04-19

Fax: (213)-2-66-26-89

## **Mots clés**

Arithmétique en ligne, transformées orthogonales, architecture pipeline.

## **Résumé**

Le processus de calcul des transformées orthogonales discrètes est caractérisé par une structure parallélo-séquentielle. Les performances obtenues pour l'exécution de ce processus sont dues essentiellement à l'exploitation de son aspect parallèle. Les possibilités de diminution des contraintes inhérentes à la nature sérielle des transformées orthogonales n'ont pas été suffisamment étudiées à ce jour. Cette communication est consacré à l'accélération de l'exécution de la partie sérielle des transformées orthogonales en utilisant, le mode half-line ( semi en ligne) et des cellules pipelines. Dans la dernière partie de la communication est présentée une méthodologie de conception de systèmes multidimensionnelles de calcul des transformées orthogonales.

## 1. INTRODUCTION

A l'exception des Transformées Optimales du type Karhunen -Loève, l'ensemble des autres transformées orthogonales, bénéficiant du développement sans précédent des techniques de conception et de fabrication des circuits VLSI, constituent aujourd'hui un outil essentiel pour le traitement du signal et de l'image. En effet, l'amélioration continue du processus de calcul des transformées orthogonales sous-optimales [1,2,3] et le développement de la technologie des circuits intégrés ont considérablement enrichi le champ d'application des méthodes de traitement du signal et d'image, notamment dans les domaines des radars, sonars, hydroacoustiques, traitement et compression de l'image et de la parole, sismologie, holographie etc...

Dans ce papier, nous proposons une approche unifiée de description du processus de calcul rapide des transformées orthogonales sous-optimales.- Fourier, Walsh-Adamard, Vilenkin-Krestenson, Haar et Cosinus,- en vue, d'une meilleure valorisation de leur nature parallélo-séquentielle et d'un élargissement de leur application à d'autres domaines.

L'approche unifiée présentée à la section 2, permet d'une part, d'utiliser des outils et des configurations arithmétiques et architecturaux adéquats avec l'exécution des tâches parallèles ou séquentielles du processus de calcul, et d'autre part, de faire le choix du ou des systèmes de fonctions orthogonales par un paramétrage des opérations de base. De ce fait, les solutions arithmétiques d'exécution de la *simultanéité*, proposées dans le présent papier, sont basées sur l'introduction, digit par digit avec poids forts en tête, de l'ensemble ou d'une partie des opérands impliqués dans l'opération de base, et l'obtention des résultats toujours digit par digit, avec un retard  $h$  inférieur à la longueur des opérands. C'est le mode on-line ou half-line de calcul [ 1,7,8,9 ] .

L'analyse comparative de calcul des transformées orthogonales en mode on-line, half-line, et parallèle ( conventionnelle ), présentée à la section 3 du présent papier fait ressortir les avantages des modes digit par digit poids fort en tête, et ce du point de vu temps de calcul, coût, et complexité.

## 2. MISE EN EVIDENCE DU CALCUL EN SIMULTANEITE ET EN PARALLELE DES TRANSFORMEES ORTHOGONALES

Les coefficients  $X(h)$  de toute transformée orthogonale d'une suite d'échantillons  $x(l)$ , donnée sur  $L$  points, constituent les éléments d'un vecteur  $\mathbf{X} = (X(0), X(1), \dots, X(h), \dots, X(L))^T$ , produit de la multiplication du vecteur  $\mathbf{x} = (x(0), x(1), \dots, x(l), \dots, x(L))^T$  par une matrice  $\Phi$  Ce qui se traduit par les expressions suivantes,

$$X = \Phi x$$

et  $x = \Phi^{-1} X$ , pour la Transformée Inverse.

Les éléments  $\phi(h,l)$  de la matrice  $\Phi$  forment un système de fonctions orthogonales et sont définies pour certaines transformées comme suit [1,6]:

Tableau 1

| Transformées  | $\phi(h,l)$   |
|---------------|---|
| Fourier       | $\text{def } (\langle h \rangle, l) = e^{-j \frac{2\pi}{L} \langle h \rangle \cdot l}$  |
| Walsh-Adamard | $\text{Had } (h,l) = (-1)^{\sum_{i=1}^n h^{(i)} \cdot l^{(i)}}$   |
| Walsh-Pelli   | $\text{pal } (\langle h \rangle, l) = (-1)^{\sum_{i=1}^n \langle h \rangle^{(n+1-i)} \cdot l^{(i)}}$  |
| Walsh         | $\text{Wal}(\omega, l) = (-1)^{\omega^{(n-i)} + \omega^{(n+1-i)}}$  |
| Haar          | $\text{Har } (h, l) = \begin{cases} 1, \text{ pour } h=0 \text{ et } l \in \{0, \dots, L-1\} \\ W^{h^{(j)} \cdot l^{(j)}}, \text{ pour } h^{(j+1)}, \dots, h^{(n-1)} = l^{(j+1)}, \dots, l^{(n-1)} \\ 0, \text{ pour } h^{(j+1)}, \dots, h^{(n-1)} = l^{(j+1)}, \dots, l^{(n-1)} \end{cases}$ |

où,

$$h = h^{(n)} \cdot 2^n + h^{(n-1)} \cdot 2^{n-1} + \dots + h^{(0)} \cdot 2^0$$

$$\langle h \rangle = h^{(0)} \cdot 2^n + h^{(1)} \cdot 2^{n-1} + \dots + h^{(n)} \cdot 2^0$$

$$\omega^{(i)} = \langle h \rangle_{\text{mod}.2}$$

$$W = e^{-j\frac{2\pi}{r}}, \text{ avec } r\text{-radix de la transformée}$$

Tout algorithme de calcul rapide des transformées orthogonales, est basé sur la décomposition de la matrice  $\Phi$  en un produit de  $\log_r L$  matrices creuses, composées chacune de  $r.L$  éléments. Outre le fait qu'elle réduit le nombre d'éléments à manipuler de  $L^2$  à  $r.L.\log_r L$ , la factorisation de la matrice de transformation permet, d'exprimer toute transformée orthogonale à travers des transformées de moindre dimension et de refléter l'organisation du processus de calcul.

A la base de la déduction des algorithmes de calcul rapide on retrouve deux outils essentiels: les matrices de permutation et le produit de Kronecker dont les définitions et les propriétés sont présentés dans [1,6].

Le processus de factorisation est un ensemble d'actions itératives de représentation des sous matrices  $\Phi_{r,m}$  à travers des sous matrices de moindre dimension,  $\Phi_{r,m-i+1}$  avec  $i = 0, 1, \dots, m-1$ . A l'étape  $i$ , chaque sous matrice est définie comme suit:

$$\Phi_{r,m-i+1} = \left\{ (h,l) \mid b \in \{0, \dots, r^{m-i+1}\}, l \in \{0, \dots, r^{m-i+1}\} \mid [b,l] = \varphi(h,l) \right\}$$

où les valeurs des éléments  $\varphi(h,l)$  sont définies par le tableau n°1. Le processus itératif se poursuivra jusqu'à ce que la matrice  $\Phi$  soit totalement exprimée à travers la matrice  $\Phi_r$ .

Ainsi pour le cas de la transformée de Walsh-Pelli, après la première itération de factorisation, la matrice de transformation prend la forme suivante:

$$H_L = (H_{L/2} \otimes I_2)(I_{L/2} \otimes H_2)$$

Au bout de la  $m$ ème itération,  $H_L$  se présente comme suit

$$H_L = (H_2 \otimes I_{2^{m-1}})(I_2 \otimes H_2 \otimes I_{2^{m-2}}) \dots (I_{2^{i-1}} \otimes H_2 \otimes I_{2^{m-i}}) \dots (I_{2^{m-1}} \otimes H_2)$$

où  $\otimes$  - symbole du produit de Kronecker  
 $I$  - La matrice identité

La mise en oeuvre de la même approche de factorisation, pour l'ensemble des transformées orthogonales sous optimales, nous conduit à une représentation recursive du processus du calcul rapide, par les expression de X suivantes

$$X = \frac{1}{L} (S_{(m)}M_{(m)} \dots (S_{(i)}M_{(i)} \dots (S_{(1)}M_{(1)}Cx) \dots) \dots) \quad (1)$$

ou,

$$X = \frac{1}{L} (CS_{(1)}M_{(1)} \dots (S_{(m-i)}M_{(m-i)} \dots (S_{(m)}M_{(m)}x) \dots) \dots) \quad (2)$$

Les matrices M, S et C sont respectivement des matrices de multiplication, de sommation (ou de liaison) et de permutation. En effet, étant une matrice diagonale, la matrice M se comporte comme un vecteur, quant à la matrice S, dont les éléments sont nuls ou égaux à 1, elle organise l'opération de sommation et de communication entre les éléments des vecteurs initial x ou intermédiaires  $X_i$ .

La structure de ces matrices pour différentes transformées est donnée dans [1]. Comme nous le verrons dans les paragraphes qui suivent, le traitement interitération ne peut se faire que dans un régime de simultanéité par l'introduction du mode on-line ou half-line de calcul. L'utilisation de ces modes nous permet de ramener le temps de calcul à  $O(\log L)$ .

### 3. CALCUL ON-LINE ET HALF-LINE DES TRANSFORMÉES ORTHOGONALES. ANALYSE COMPARATIVE.

L'opérateur généralisé constitue un système d'expressions algébriques de même type, appelées "procédure élémentaire" et dont le nombre et la complexité en terme d'opérations dépendent du type de transformée orthogonale et de la valeur du radix r. Une procédure élémentaire est de la forme:

$$\sum_{e=1}^z \prod_{j=1}^{l(e)} * A_{e,j} \quad (3)$$

où le symbole \* est égal à  $\pm 1$ . Les valeurs de z et de l(e), ainsi que le nombre u de procédures élémentaires pour chaque transformée sont portés dans les tableaux suivants:

Tableau 2

|   |     |                              |     |     |                         |     |
|---|-----|------------------------------|-----|-----|-------------------------|-----|
|   | $z$ | $l(e)$                       | $u$ | $z$ | $l(e)$                  | $u$ |
| 2 |     | $l(1)=1$<br>$l(2)=l(3)=2$    | 4   | 2   | $l(1)=1$<br>$l(2)=1$    | 2   |
| 4 | 7   | $l(1)=1$<br>$l(2)=..=l(7)=2$ | 8   | 4   | $l(1)=1$<br>$..=l(4)=1$ | 4   |

Tableau 3

| $r$ | FHT |                             |     | FVKT |   |     |
|-----|-----|-----------------------------|-----|------|---|-----|
|     | $z$ | $l(e)$                      | $u$ | $z$  | $l(e)$  | $u$ |
| 2   | 2   | $l(1)=l(2)=1$               | 2   | 2    | $l(1)=l(2)=1$                                   | 4   |
| 4   | 4   | $l(1)=.....$<br>$..=l(4)=1$ | 4   | 4    | $l(1)=1$<br>$..=l(4)=1$                         | 8   |
| 8   | 8   | $l(1)=.....$<br>$=l(8)=1$   | 8   | 12   | $l(1)=...=1$<br>$(4)=1, l(5)=1$<br>$..=l(12)=2$ | 16  |
| 8   | 15  | $l(1)=1$<br>$=l(8)=1$       | 8   | 12   | idem  | 16  |

Le calcul de la procédure élémentaire en mode parallèle ou série avec poids faibles en tête, par la méthode divide and conquer s'effectue en  $(l(e)-1)$  étapes de multiplications et  $\log z$  étapes d'additions. Dans chacune de ces étapes les opérations de multiplication et d'addition se font en parallèle. Ainsi le résultat de la procédure ne peut être obtenu qu'après un nombre  $(l(e)-1) \cdot \log L + \log z$  étapes séquentielles d'additions, si on considère que  $\log L$  est la taille des opérandes manipulés [10]. Par ailleurs, la procédure élémentaire suivante ne peut être entamée avant que les procédures qui la précèdent et dont elle dépend soient totalement exécutées. Ceci est dû à la récursivité du processus de calcul. Ces actions séquentielles aussi bien au niveau de la procédure élémentaire qu'au niveau de tout le processus de calcul des transformées représentent une contrainte primordiale dans l'accroissement de l'accélération des transformées orthogonales. Pour remédier à cette contrainte, nous proposons l'utilisation des modes on-line et half-line

#### CALCUL DES TRANSFORMÉES EN MODE ON-LINE

Le mode on-line est défini par les propriétés suivantes :

a- Les opérandes sont introduits dans l'opérateur, à chaque cycle, digits par digits avec poids forts en tête. Autrement dit le lancement de l'opération ne nécessite la connaissance que d'un digit de chaque opérande.

b- Les digits du résultat sont eux aussi obtenus, digit par digit, mais avec un retard  $h$ , tel qu'au pas  $j$ , alors que les  $j^{\text{ième}}$  digits des opérandes ont été introduits, le  $(j-h)^{\text{ième}}$  digit du résultat est généré.

c- Le retard  $h$  est très petit devant la longueur des opérandes.

d- Le résultat et éventuellement les opérandes, sont présentés dans un système d'écriture redondant.

L'objectif essentiel lors de l'exécution de ce mode de calcul demeure la réduction du délai  $h$  de génération du premier digit résultat

Le calcul on-line de la procédure élémentaire est exécuté par une cellule élémentaire construite, soit à base de binomieurs, soit à base d'opérateurs plus importants que nous étudierons en détail plus loin.

La première approche nécessiterait l'utilisation de  $(z-1)$  binomieurs, ayant chacun un délai de génération du premier digit résultat de  $h=3$  cycles [11]. Pour le cas de de la FFT, radix  $r=4$ , l'expression de la procédure élémentaire serait de la forme suivante:

$$(A_{11} + *A_{21}A_{22}) + *A_{31}A_{32} + *A_{41}A_{42}.....) + *A_{71}A_{72}$$

Dans ce cas, la cellule élémentaire aura la configuration suivante :

Le délai de génération du premier digit-résultat de la procédure élémentaire, sera de  $18 t_c$ , où  $t_c$  est la période d'introduction de deux digits successifs dans la cellule élémentaire. Cela signifie que le premier digit-résultat sera obtenu qu'après l'introduction des 18-ièmes digits des opérandes  $A_{21}$  et  $A_{11}$ . Ce fait réduit considérablement les avantages du mode on-line, qui se comporte, alors, comme un calcul en bit-série avec poids faibles en tête. Par ailleurs, l'introduction en asynchrone des opérandes complique la gestion de la mémoire et le contrôle de la cellule élémentaire.

Pour remédier à toutes ces contraintes, nous proposons le design d'une cellule élémentaire spécifique et destinée à l'exécution, exclusivement, de la procédure élémentaire et des processus construits à la base de cette procédure.

Exprimons à travers

$$A_{e,j}^{(q)} = \{a_{e,j}^{(0)}, a_{e,j}^{(1)}, a_{e,j}^{(2)}, \dots, a_{e,j}^{(q)}\} \quad (4)$$

et

$$S^{(q)} = \{s^{(0)}, s^{(1)}, s^{(2)}, \dots, s^{(q)}\} \quad (5)$$

respectivement l'ensemble des opérandes et des résultats partiels, où  $a^{(q)}$  et  $s^{(q)}$  sont les digits de poids  $q$ , qui prennent leurs valeurs dans l'ensemble  $\{-1, 0, 1\}$ .

Le problème consiste alors, à définir la valeur minimale du délai  $h$ , pour laquelle le processus de calcul de  $S^{(q)}$  converge vers  $S$ , et à élaborer l'algorithme de génération des digits-résultats  $s^{(q)}$  .. au fur et à mesure de l'apparition des digit-opérandes  $a^{(q)}$ .

Pour ce faire, il est nécessaire de noter, qu'à chaque  $q$ -ième pas de calcul, soit accomplie la condition d'arrondissement suivante:

$$S^{(q)} - 2^{-(q+1)} \leq \sum_{e=1}^z \prod_{j=1}^{l(e)} A_{e,j}^{(q)} / D < S^{(q)} + 2^{-(q+1)} \quad (6)$$

et par conséquent, après l'exécution de  $(n+1)$  cycles, sera obtenu  $S^{(n)} = S$  avec une erreur non supérieure à  $|2^{-(n+1)}|$

En distribuant l'expression (8), nous obtenons :

$$-D/2 \leq \left[ \sum_{e=1}^z \prod_{j=1}^{l(e)} A_{e,j}^{(q)} - S^{(q)} \cdot D \right] \cdot 2^q < D/2 \quad (7)$$

L'expression du milieu représente le reste  $R_0^{(q)}$  de la division de  $\sum_{e=1}^z \prod_{j=1}^{l(e)} A_{e,j}^{(q)}$  par  $D$ ,

$$R_0^{(q)} = \left[ \sum_{e=1}^z \prod_{j=1}^{l(e)} A_{e,j}^{(q)} - S^{(q)} \cdot D \right] \cdot 2^q \quad (8)$$

En convenant que:

$$A_{e,j}^{(q)} = A_{e,j}^{(q-1)} + a_{e,j}^{(q)} \cdot 2^{-q} \quad \text{et} \quad S^{(q)} = S^{(q-1)} + s^{(q)} \cdot 2^{-q}$$

alors,



$$R_0^{(q)} = \sum_{e=1}^Z \prod_{j=1}^{l(e)} [(A_{e,j}^{(q-1)} + a_{e,j}^{(q)} \cdot 2^{-q}) - (S^{(q-1)} + s^{(q)} \cdot 2^{-q}) \cdot D] \cdot 2^q$$

Cette dernière relation exprime le reste  $R_0^{(q)}$  à travers, les valeurs des opérandes tronqués  $A_{e,j}^{(q-1)}$ , les résultats partiels  $S^{(q-1)}$ , les digits-résultats  $s^{(q)}$  et les digits-opérandes  $a_{e,j}^{(q)}$ . Sachant que la division est un processus itératif, il est utile d'exprimer  $R_0^{(q)}$  à travers  $R_0^{(q-1)}$ .

A cet effet, on effectue la distribution de l'expression  $R_0^{(q)}$  en tenant compte du fait que  $l(e) \in \{1, 2, 3\}$ .

En prenant en considération la formule de récurrence (8), on obtient:

$$\begin{aligned} R_0^{(q)} = & 2R_0^{(q-1)} + \sum_{e=1}^Z (a_{e,1}^{(q)} A_{e,2}^{(q-1)} \dots A_{e,l(e)}^{(q-1)} + A_{e,1}^{(q-1)} a_{e,2}^{(q)} \\ & \dots A_{e,l(e)}^{(q-1)} + A_{e,1}^{(q-1)} A_{e,2}^{(q-1)} \dots a_{e,l(e)}^{(q)} + a_{e,1}^{(q)} a_{e,2}^{(q)} \dots A_{e,l(e)}^{(q-1)} \cdot 2^{-q} \\ & + a_{e,1}^{(q)} A_{e,2}^{(q-1)} \dots a_{e,l(e)}^{(q)} \cdot 2^{-q} + A_{e,1}^{(q-1)} a_{e,2}^{(q)} \dots a_{e,l(e)}^{(q)} \cdot 2^{-q} + \\ & a_{e,1}^{(q)} a_{e,2}^{(q)} \dots a_{e,l(e)}^{(q)} \cdot 2^{-q} \cdot 2^{-q}) - s^{(q)} \cdot D \end{aligned} \quad (9)$$

En désignant l'expression sous la somme à travers  $\xi_{(e)}^{(q)}$ , on obtient les relations récurentes suivantes:

$$\begin{aligned} R^{(q)} = & 2 R^{(q-1)} + 2^{-h} \cdot \sum_{e=1}^Z \xi_{(e)}^{(q)} - s^{(q)} \quad \text{et} \\ H^{(q)} = & 2 R^{(q-1)} + 2^{-h} \cdot \sum_{e=1}^Z \xi_{(e)}^{(q)} \end{aligned} \quad (10)$$

avec,  $R^{(q)} = R_0^{(q)} \cdot 2^{-h}$ ,  $H^{(q)} = H_0^{(q)} \cdot 2^{-h}$  et  $h = \log D$ .

Ainsi, à chaque cycle  $q > h$  seront introduits les digit-opérandes  $a_{e,j}^{(q)}$  de poids  $2^{-q}$  et seront générés les digit-résultats de poids  $2^{-(q+h)}$ . En vue de diminuer le retard de génération du premier digit-résultat, il est nécessaire de choisir la valeur minimale de  $h$ , qui se présente comme suit:

$$h = \log_2 \sum_{e=1}^z k(\xi_e^{(q)}) + 1 \quad (11)$$

où  $k(\xi_{(e)}^{(q)}) = \left( \sum_{e=1}^z \xi_{(e)}^{(q)} \right)_{\max}$

Quant aux critères de choix du digit-résultat ils résultent de la condition d'arrondissement (7) qui, en tenant compte de la relation (10), prend la forme suivante:

$$s^{(q)} - 1/2 \leq H^{(q)} < s^{(q)} + 1/2 \quad (12)$$

En affectant à  $s^{(q)}$ , les différentes valeurs de l'ensemble  $\{-1, 0, 1\}$ , ces critères se présentent comme suit:

$$s^{(q)} = \begin{cases} -1, & -3/2 \leq H^{(q)} < -1/2 \\ 0, & -1/2 \leq H^{(q)} < 1/2 \\ 1, & 1/2 \leq H^{(q)} < 3/2 \end{cases} \quad (13)$$

Par conséquent, le choix de  $s^{(q)}$  pour chaque cycle, se fait en deux étapes:

Première étape

.Calcul de H par le biais de l'expression

$$H^{(q)} = 2 R^{(q-1)} + 2^{-h} \cdot \sum_{e=1}^z \xi_{(e)}^{(q)}$$

Deuxième étape

.Analyse des trois digits de H, ayant pour poids  $2^{-1}$ ,  $2^0$  et  $2^1$  en vue de définir, à quelle intervalle de l'expression (15) appartient H. Génération de  $s^{(q)}$ .

#### ESTIMATION DU TEMPS DE CALCUL

Le coût d'exécution de cet algorithme est estimé, en nombre d'additions de termes simples, contenus dans l'expression de H. Par terme simple on désignera un opérande ou le produit d'un opérande par un digit. Comme

illustré par l'expression (9) le nombre  $P(e)$  de termes simples inclus dans  $\xi(e)$ , dépend des valeurs de  $l(e)$ .

Ainsi, pour  $l(e)=1,2$  ou  $3$ ,  $P(e)$  sera égal respectivement à  $1,3$ , et  $16$ . Par conséquent, le nombre total d'opérandes simples contenus dans  $\sum_{e=1}^z \xi(e)^{(g)}$  est défini

comme suit:  $\tilde{P} = \sum_{e=1}^z P(e)$

En considérant la dernière expression et les relations (10) on peut déduire que pour chaque cycle de génération du digit-résultat, il est nécessaire d'effectuer  $P = \tilde{P} + 2$  additions. Dans le tableau 4 sont présentées les valeurs de  $P$  et de  $h$  des transformées de Fourier, Walsh, Haar, de Vilenkin-Krestenson pour les radix  $2, 4$ , et  $8$ .

Tableau 4

|            | h | P | h | P  | h | P  |
|------------|---|---|---|----|---|----|
| FFT        | 4 | 9 | 5 | 21 | 6 | 45 |
| Walsh      | 2 | 4 | 3 | 6  | 4 | 10 |
| Haar       | 2 | 4 | 3 | 6  | 4 | 10 |
| Vilen-Kres | 2 | 4 | 3 | 6  | 6 | 30 |

L'examen de ce tableau fait ressortir le fait suivant: Comparativement à l'utilisation des binomieurs, le délai de génération du premier digit - résultat est de trois fois moins important, ce qui justifie notre choix pour l'évaluation de la procédure élémentaire par une cellule élémentaire spécifique. L'utilisation de la méthode divide and conquer et l'assimilation de l'analyse des trois digits à une opération d'addition, permettent d'accomplir tout le cycle ( calcul de  $H$  et analyse des digits MSB) en:  $\lambda = \log P + 1$  additions.

Définissons un opérateur élémentaire comme étant un module, constitué de  $u$  cellules élémentaires. Les valeurs de  $u$  sont données dans les tableaux 2 et 3. Dans ce cas, pour une architecture du type network, constituée d'opérateurs élémentaires en on-line, le temps d'exécution d'une transformée orthogonale se présente comme suit:

$$T = [ (h + 1) \cdot \log P + 1 ] \cdot \log L + (n - 1) \cdot (\log P + 1) \cdot \tau. \quad (14)$$

En posant  $n = c \cdot \log L$ ,

$$T = [((h + c + 1) \cdot \log P + c + 1) \log L - (\log P + 1)] \cdot \tau_+ \quad (15)$$

A partir de cette expression et des données du tableau 4 sont tracées les courbes des temps de calcul des différentes transformées et ce pour  $c=1$ . A la lecture de ces courbes (fig.1) et comme il fallait s'y attendre, le temps d'exécution le plus faible est celui de la transformée de Walsh radix 8, suivent ensuite les transformées de Walsh radix 2 et 4, de Vilenkin-Krestenson, et enfin les FFT radix 8, 4 et 2. Il est à remarquer que les FFT radix 4 et 8 s'exécutent en des temps relativement rapprochés.

Selon l'expression (17), l'accroissement des performances de calcul des transformées orthogonales peut être obtenu d'une part, par une diminution du délai  $h$  et du nombre de termes simples  $P$  et d'autre part, par une amélioration du temps  $\tau_+$  d'exécution de l'addition et par l'accélération du calcul des  $\log P$  étages d'additions de termes simples. Ces conditions peuvent être obtenues d'une part par l'utilisation du mode half-line et d'autre part par la mise en oeuvre d'une architecture pipeline au niveau de la cellule élémentaire.

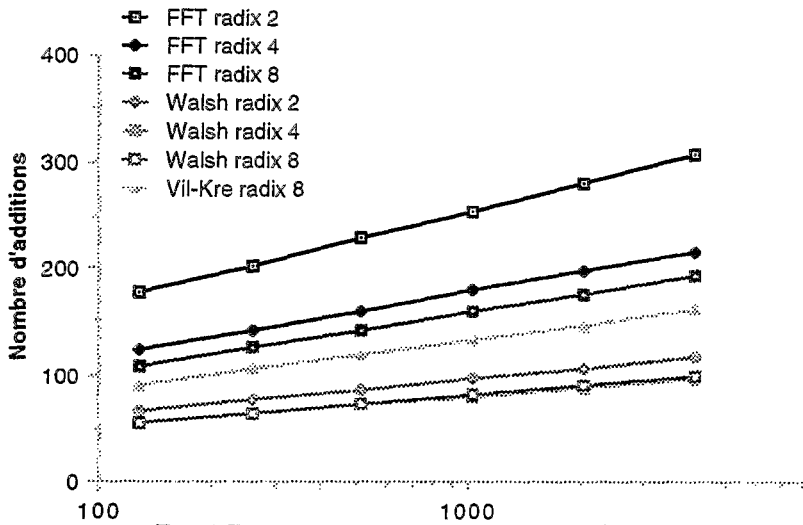


Fig. 1 Temps de calcul en ligne des TRDO

Il est aisé de montrer [12], que dans le mode half-line les valeurs de h et de P peuvent être calculés par les expressions suivantes:

$$h = \overline{\log z + 1}$$

et

$$P = z + 2$$

De ces expressions et des valeurs de z, définies dans les tableaux 3 et 4, découlent les délais et le nombre de termes simples pour les transformées orthogonales, objets de cette étude. Les résultats sont portés dans le tableau ci-dessous

Tableau 5

|            | h | p | h | p | h | p  |
|------------|---|---|---|---|---|----|
| FFT        | 3 | 5 | 4 | 9 | 5 | 17 |
| Walsh      | 2 | 4 | 3 | 6 | 4 | 10 |
| Haar       | 2 | 4 | 3 | 6 | 4 | 10 |
| Vilen-Kres | 2 | 4 | 3 | 6 | 6 | 28 |

Un bref examen des valeurs de h et de P, présentées dans les tableaux 5 et 6 fait ressortir un avantage indiscutable du mode half-line, par rapport au mode on-line, en particulier lors du calcul de la transformée de Fourier.

Des expressions (10) H(q) peut s'écrire sous la forme suivante:

$$H^{(q)} = 2H^{(q-1)} + 2^{-h} \sum_{e=1}^z \xi_{(e)}^{(q)} - 2s^{(q-1)} \quad (16)$$

Le processus pipeline, de calcul de cette expression, est l'un des moyens les plus efficaces d'exécution d'une somme de termes simples, organisée sous forme d'un arbre, particulièrement dans un environnement séquentiel.

La fig.2 illustre le fonctionnement d'une cellule élémentaire pipeline.

Les places ( ou positions ) représentent les registres interclaires ( banc de bascules ) du pipeline, les transitions - les parties combinatoires.

Le fonctionnement de la cellule élémentaire est le suivant:

Lors de la première étape, illustrée par les places P1-P2l' et les transitions T1-

T2l', est effectuée l'opération  $2^{-h} \sum_{e=1}^z \xi_{(e)}^{(q)}$ . Chaque transition est constituée d'une rangée d'additionneurs. L'ensemble des transitions T1 - T2l' sont organisées en arbre et à l'entrée de chacune d'elle, est disposée une position. Le couple

position-transition constitue un étage du pipeline. Il est évident que le nombre d'étages du pipeline dans cette partie de la cellule élémentaire dépend du type de transformée et de la valeur du radix.

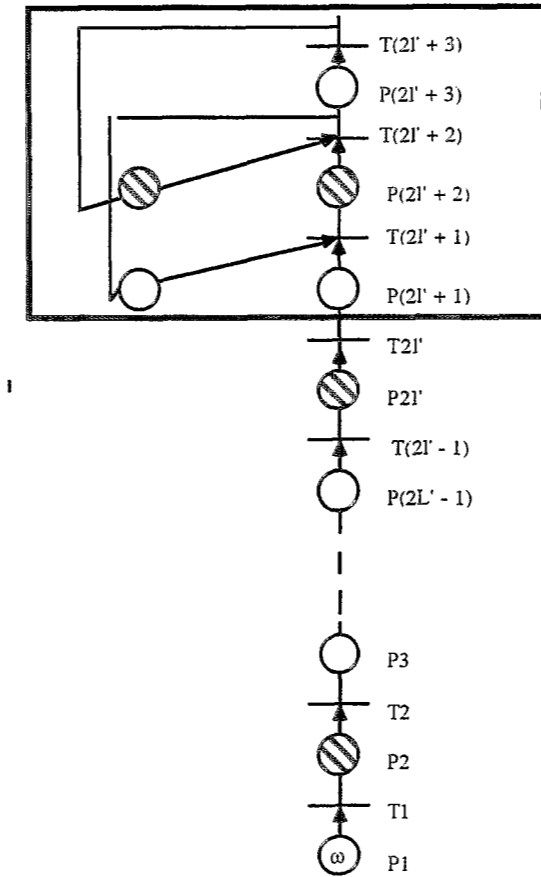


fig.2 Cellule élémentaire pipeline

La deuxième étape du processus de génération du digit-résultat, matérialisé par la tête de la cellule (la partie encadrée de la fig.5), et constituée des positions, P1'

+ 1 ÷ P'2l' + 3; P'2l' + 1; P'2l' + 2 et des transitions T2l' + 1 ÷ T2l' + 3, réalise la somme des termes simples suivants:  $2H^{(q-1)}$ ,  $-2s^{(q-1)}$  et  $2^{-h} \sum_{e=1}^z \xi_{(e)}^{(q)}$ . Les fonctions des positions et des transitions ne dépendent pas de la structure de l'opérateur généralisé. Elles se présentent come suit:

$$[ P'2l' + 1 ] = 2^{-h} \sum_{e=1}^z \xi_{(e)}^{(q)}$$

$$[ P'2l' + 2 ] = 2H^{(q-1)}$$

$$[ P'2l' + 2 ] = 2H^{(q-1)} + 2^{-h} \sum_{e=1}^z \xi_{(e)}^{(q)}$$

$$[ P'2l' + 2 ] = 2s^{(q-1)}$$

$$[ P'2l' + 3 ] = H^{(q)}$$

Où [a] signifie le contenu de la position a.

Les transitions T(2l' + 1) et T(2l' + 2) réalisent la somme de deux nombres de taille (n+h+1). Quant à la dernière transition T(2l'+3), elle sert au décodage des trois digits les plus forts de  $H^{(q)}$ , donc à la génération du digit-résultat, lequel est transféré à la cellule suivante et, à travers une boucle, à la position P'(2l' + 2) de la même cellule.

Le temps de calcul peut considérablement diminuer par l'introduction de carry save adder sans propagation de la retenue au niveau des étages du pipeline, en utilisant des additionneurs à conservation de retenu (CSA). Sachant que la valeur de  $H^{(q)}$  sert essentiellement à générer les digit-résultats, le problème se pose en terme de nombre minimal de digits de  $H^{(q)}$ , permettant de déterminer l'appartenance de ce dernier à l'un des intervalles, définis par les expressions (13).

L'utilisation de la méthode CSA suppose l'obtention à chaque étape de calcul d'une demi somme et de la retenue et ce, grace à un additionneur à conservation de retenue, constitué de n additionneurs complets.

Dans ce cas le temps de calcul d'une transformée serait

$$T = [(1/3) \cdot \log_{3/2} 2\tilde{P} + h + 1] \cdot \log_r L + n - 1] \cdot 4\tau_+^{(1)}$$

ou

$$T = [(1/3 \cdot \log_{3/2} 2\tilde{P} + h + c + 1) \cdot \log_r L - 1] \cdot 4\tau_+^{(1)}$$

#### 4. Conclusion

Le processus de calcul des transformées orthogonales est caractérisé par une structure parallélo-séquentielle.

Nous avons montré dans ce papier, que l'accélération du processus de calcul des transformées orthogonales peut s'effectuer par l'utilisation d'une arithmétique et d'une microarchitecture adéquates, en l'occurrence les modes on-line et half-line de calcul, et une architecture pipeline avec un carry save adder au niveau des étages.

Les algorithmes élaborés, les estimations effectuées et les analyses comparatives entreprises, permettent de conclure que le calcul halfline des transformées orthogonales radix 4, implémentées sur une architecture network, construite à base d'opérateurs pipelines élémentaires avec trois horloges de synchronisation, est le plus performants du point de vue temps de calcul, qui est estimé à  $O(\log L)$ .

La complexité et la faisabilité d'un système network est lié à la maîtrise du nombre d'opérateurs élémentaires, à la réduction du volume des données circulant entre ces opérateurs et à la gestion synchrone de l'acheminement de ces données. L'utilisation du mode half-line et de l'architecture pipeline permettent de répondre à ces conditions, notamment aux deux dernières. Quant à la première condition, elle peut être remplie grâce à l'utilisation de processeurs élémentaires VLSI, composés de plusieurs opérateurs élémentaires et, effectuant chacun une transformée orthogonale d'un certain nombre d'échantillons. La synthèse du système network et des processeurs élémentaires, qui le constituent font l'objet de la deuxième partie du présent travail.

#### References

1. H. Bessalah, Multidimensional pipeline implementation for on-line computation of fast orthogonal transform, Ph. D Thesis. Polytechnical Institut of Kiev, 1981.