

MISE EN ŒUVRE D'UN NOUVEL ALGORITHME DE
TRANSFORMATION DE FOURIER SUR PROCESSEUR
SPECIALISE

Gay TALL
Richard GRISEL

I.N.S.A. de ROUEN
LCIA ITEPEA
B.P. 08 76131 Mont Saint Aignan Cedex

Résumé. L'utilisation des processeurs de traitement du signal pour le calcul des algorithmes à transformation de Fourier permet un gain de temps considérable. Cet article, présente l'algorithme à base double sous une forme adaptée au calcul par processeur spécialisé. Une étude de programme pour l'ADSP-2100 d' Analog Devices est abordée. Les avantages d'utiliser un tel algorithme sont montrés et une comparaison est faite avec les algorithmes classiques de Cooley et Tuckey.

Abstract. The use of digital signal processing to compute Fourier transformation algorithms allows a great saving of time. This paper presents the split radix algorithm under a form which is adapted to calculation by specialized processor. We begin a study of program for the analog Devices ADSP-2100 microprocessor in this paper. We show the advantages linked to the use of such an algorithm and we make a comparison with the classical Cooley's and Tuckey's.

1. INTRODUCTION

Le calcul direct de la transformée de Fourier discrète nécessite quelques millions de multiplications et additions réelles. Ce nombre peut être considérablement réduit en utilisant des algorithmes dits a "Transformée de Fourier Rapide" appelés couramment FFT (Fast Fourier Transform). Plusieurs algorithmes ont été proposés :

- Cooley et Tuckey (1963)
- Rader et Brenner (1976)
- Winograd (1976)
- Split Radix (1985)

Cependant, malgré les performances exceptionnelles de ces algorithmes l'obtention des résultats en "temps réel" se révèle souvent impossible en pratique. Le recours aux processeurs spécialisés s'avère obligatoire dans ce cas. La réalisation concrète d'un analyseur de spectre revient alors à matérialiser l'algorithme. De ce fait, l'interaction entre algorithme et architecture du processeur devient une donnée cruciale à prendre en compte avant toute autre considération.

Les algorithmes récents sont d'un point de vue théorique plus rapide que celui de Cooley et Tuckey mais en pratique celui-ci reste le plus utilisé. En effet toutes les architectures existantes sont calquées sur ce modèle. La complexité des algorithmes récents explique aussi cet fait.

Partant de cette constatation, il nous a paru opportun de présenter une forme du "split radix" qui se prête bien au calcul sur processeur spécialisé.

Nous proposons un code pour l'ADSP-2100 d'Analog Devices, lequel possède une architecture interne qui le destine particulièrement à l'exécution de l'algorithme de Cooley et Tuckey. Nous terminons cet article en comparant les temps avec ceux des algorithmes classiques.

2. ALGORITHME A BASE DOUBLE

La transformée de Fourier discrète est définie par :

$$(1) \quad X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad \text{pour } k=0,1,\dots,N-1 \text{ et}$$

$W_N^{nk} = \exp\left(\frac{2\pi nk}{N}\right)$ nécessite N^2 multiplications complexes et N^2 additions complexes. L'algorithme à base double divise d'abord la série d'entrée $x(n)$ en deux sous-groupes de $\frac{N}{2}$ points. Le second sous-groupe est de nouveau divisé en deux séquences de $\frac{N}{4}$ points. L'algorithme réalise alors une TFD de $\frac{N}{2}$ points et deux TFD de $\frac{N}{4}$ points. Cette première fragmentation réduit le nombre d'opérations de :

$$\left(\frac{N}{2}\right)^2 + \left(\frac{N}{4}\right)^2 + \left(\frac{N}{4}\right)^2 = 0,375N^2 \text{ soit un gain de } 62,5\%.$$

D'autres réductions sont ainsi accomplies en répétant successivement cette procédure sur de plus petites TFD.



En dédoublement temporel, la séquence d'entrée -temporelle- est d'abord divisée en deux groupes l'un est constitué des échantillons d'indices pairs et l'autre des échantillons d'indices impairs. Ensuite la séquence des échantillons d'indices impairs est de nouveau divisée en sous-séquences paires et impaires. Cela revient à appliquer une base 2 pour les termes pairs et une base 4 pour les termes impairs.

Trois transformées, une de $\frac{N}{2}$ points et deux de $\frac{N}{4}$ points, sont calculées sur ces nouvelles séquences. L'équation (1) devient

$$(2) \quad X(k) = \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + \sum_{n=0}^{N/4-1} x(4n+1)W_N^{(4n+1)k} + \sum_{n=0}^{N/2-1} x(4n+3)W_N^{(4n+3)k}$$

Calculer une TFD de taille N revient à calculer une TFD de taille $\frac{N}{2}$ et deux TFD de taille $\frac{N}{4}$.

En remarquant que :

$$W_N^{2nk} = W_{N/2}^{nk}$$

$$W_N^{(4n+1)k} = W_N^k W_{N/4}^{nk}$$

$$W_N^{(4n+3)k} = W_N^{3k} W_{N/4}^{nk}$$

Cette équation peut se réécrire :

$$(3) \quad X(k) = X_1(k) + W_N^k X_2(k) + W_N^{3k} X_3(k)$$

Elle est la somme de trois TFD. $X_1(k)$ est une TFD à $\frac{N}{2}$ points $X_2(k)$ et $X_3(k)$ sont deux transformées à $\frac{N}{4}$ points et sont respectivement multipliées par les facteurs de rotation W_N^k et W_N^{3k} . Chacune de ces trois transformées est calculée en utilisant la décomposition ci-dessus. Cette procédure est répétée jusqu'à ce que les séquences se réduisent à deux points. La figure 1 explicite les dédoublements fréquentiels successifs de la série d'entrée.

Par définition, les séries $X_1(k)$, $X_2(k)$ et $X_3(k)$ sont périodiques. On a donc :

$$X_1(k + \frac{N}{2}) = X_1(k),$$

$$X_2(k + \frac{N}{4}) = X_2(k) \text{ et}$$

$$X_3(k + \frac{N}{4}) = X_3(k).$$

On peut tirer profit de ces périodicités en calculant l'équation (3) avec la condition ($k < \frac{N}{4}$). Ce qui nous donne les équations suivantes définissant l'opération élémentaire à effectuer en base double.

$$(4) \quad X(k) = X_1(k) + W_N^k X_2(k) + W_N^{3k} X_3(k)$$

$$(5) \quad X(k + \frac{N}{4}) = X_1(k + \frac{N}{4}) - j W_N^k X_2(k) + j W_N^{3k} X_3(k)$$

$$(6) \quad X(k + \frac{N}{2}) = X_1(k) - W_N^k X_2(k) - W_N^{3k} X_3(k)$$

$$(7) \quad X(k + \frac{3N}{4}) = X_1(k + \frac{N}{4}) + j W_N^k X_2(k) - j W_N^{3k} X_3(k),$$

pour $k=0,1,\dots, \frac{N}{4}-1$.

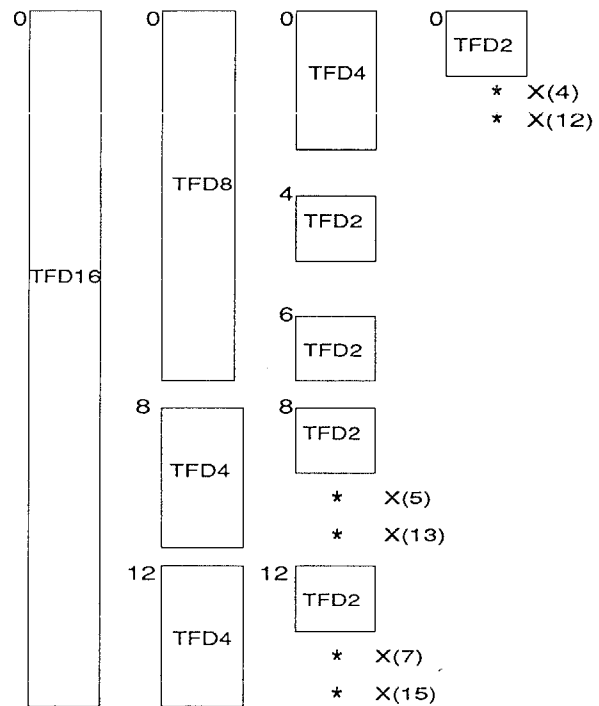


Figure 1: Dédoublement fréquentiel

2.1 Les papillons

La figure 2 représente la cellule élémentaire de l'algorithme à base double. Quelques papillons en base 2 sont nécessaires pour compléter le calcul. Les papillons d'ordre 4 et 2 ne nécessitant aucune multiplication seront calculés séparément. Les équations de sortie pour le papillon général papillon sont données ci-dessous :

$$r1 = x(n) - x(n+N/2), \quad x(n) = x(n) + x(n+N/2)$$

$$i1 = y(n) - y(n+N/2), \quad y(n) = y(n) + y(n+N/2)$$

$$r2 = x(n+N/4) - x(n+3N/4), \quad x(n+N/4) = x(n+N/4) + x(n+3N/4)$$

$$i2 = y(n+N/4) - y(n+3N/4), \quad y(n+N/4) = y(n+N/4) + y(n+3N/4)$$

$$A = i1 + r2, \quad B = i1 - r2, \quad C = r1 + i2, \quad D = r1 - i2$$

$$x(n+N/2) = C * TW1C[i] - B * TW1S[i]$$

$$y(n+N/2) = C * TW1C[i] + B * TW1S[i]$$

$$x(n+3N/4) = D * TW3C[i] - A * TW3S[i]$$

$$y(n+3N/4) = D * TW3S[i] + A * TW3C[i]$$

Ici, N représente la taille du groupe. TW1C, TW1S, TW3C et TW3S sont les tableaux qui contiennent les parties réelles et imaginaires des facteurs de rotation W_N^k et W_N^{3k} .

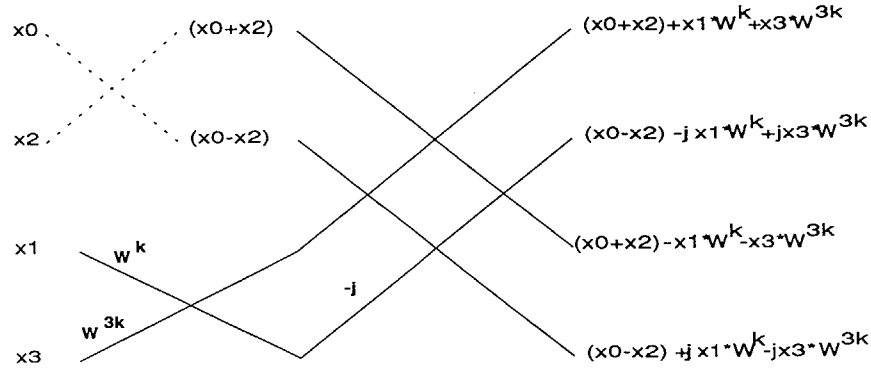


Figure2 : Papillon général du " split radix " dédoublement temporel sur quatre points .Un papillon base 2 est nécessaire pour compléter le calcul.

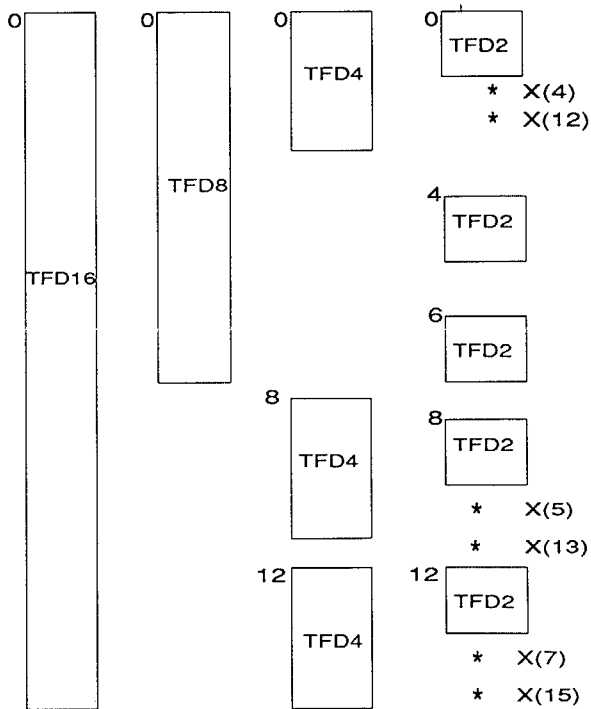


Figure3 :Dédoublment fréquentiel réorganisé

3. ETUDE DU PROGRAMME

La décomposition de l'algorithme à base double représentée sur la figure 1 fait apparaître une irrégularité dans l'organisation des étapes. On retrouve en effet dans une étape des groupes de papillons de taille différente ce qui complique la programmation sur DSP. Cette difficulté peut être contournée en utilisant la structure de la figure 3 où les groupes sont réorganisés de telle sorte que chaque étape ne comporte que des groupes de taille identique.

La seconde difficulté liée à cette décomposition est l'ordre d'apparition des groupes dans les étapes. De la façon qu'ils se présentent le calcul des indices de début des blocs n'est pas simple. Pour éviter cette difficulté, nous proposons de calculer au préalable tous les indices de début des groupes et de les stocker dans un tableau " INDICES ". Ensuite, il suffira de consulter ce tableau pour trouver les indices.

En dehors de ces deux difficultés la structure du "split radix " est d'une souplesse comparable à ceux de Cooley et Tuckey et se programme de la même façon.

Ce programme est constitué de quatre parties. La première partie initialise les variables (taille des blocs,écarts entre noeuds duals, modificateur d'adresse des pointeurs ...). La partie suivante calcule toutes les étapes comportant des papillons de type général. La troisième section exécute les groupes à quatre points de l'avant-dernière étape. La dernière partie effectue le calcul des papillons d'ordre deux de l'étape finale. L'organisation détaillée de ce programme est décrite ci-après. Le code correspondant peut être demandé au laboratoire.

Initialisation :

```
{ N=1024 , M=10,M_2=8, N2=N/2, stdw=1
  N4=N2/2,3n4=N2+N/4,NB=1,NB_1=0,NB_2=0,i=0}
```

Etape à papillons généraux :

```
De e=1 à M-2
  { de b=1 à NB ;tous les blocs de l'étape
    { ib=indices[i]
      itdw=0; indice des facteurs de rotation
      de p=1 à N4
        { Calcul groupe à papillons
          généraux ;
          i=i+1 }
      N2=N2/2, N4=N4/2, 3N4=3N4/2
      stwd=stwd*2
      NB_2=NB_1
      NB_1=NB
      NB=NB_1+2*NB_2
    }
  }
```

Etape groupe à quatre points :

```
De b=1 à NB { calcul groupe à 4 points
  i=i+1 }
NB=NB+2*NB_1
```

Etape papillons à deux points :

```
De b=1 à NB { calcul papillon à deux points
  i=i+1 }
```



4. PERFORMANCES ET CONCLUSION

Les algorithmes et les processeurs qui permettent de calculer la transformée de Fourier évoluent sans cesse mais leur mise en relation n'est toujours pas évidente. Nous avons proposé dans cet article un exemple d'interaction possible entre processeur et algorithme. Sur un processeur donné, les performances d'une FFT vont dépendre essentiellement de l'algorithme utilisé. A titre indicatif, le programme du "split radix" tel que nous l'avons réalisé sur l'ASP-2100 effectue une FFT sur 1024 points en 4 ms. Ce résultat est comparable au temps obtenu en base 4 - 3 ms sur 1024 - et il est supérieur à celui de la base 2 (6 ms sur 1024points).

BIBLIOGRAPHIE

1. J.LFFERMAN, Les méthodes rapides de la transformation du signal, Masson,1974.
2. P. DUHAMEL,H. HOLMANN, Split-radix FFT algorithm, Electron.lett., B, 20, numéro 1, pp 14-16.
3. P. DUHAMEL,Un algorithme de transformation de Fourier rapide à base double, Annales des télécommunications,4, numéro 9-10,1985, p 481-494.
4. ADSP-2100 FAMILY,Applications Handbook,volume 1.