

Implantation d'un opérateur reconfigurable de Transformée de Fourier Rapide (TFR)

Florent CAMARDA, Jean-christophe PREVOTET, Fabienne NOUVEL

Institut d'Electronique et de Télécommunications de Rennes
20 avenue des buttes de coëmes 35043 Rennes

florent.camarda@insa-rennes.fr, jean-christophe.prevotet@insa-rennes.fr,
fabienne.nouvel@insa-rennes.fr

Résumé – Cet article présente une nouvelle architecture pour le traitement de la Transformée de Fourier Rapide (TFR) de taille variable. Après avoir introduit le contexte de la télévision numérique et présenté les avantages liés aux aspects de reconfigurabilité des architectures pour des applications multistandards, les algorithmes Radix et WFTA exploités par l'architecture proposée sont présentés. L'architecture est par la suite décrite avant d'en présenter quelques performances.

Abstract – This paper present a new reconfigurable architecture allowing a variable length Fast Fourier Transform computation. After having introduced the digital TV broadcasting context and presented the benefits of reconfigurable architectures for multistandards applications, The used Radix and WFTA algorithms are presented. The proposed architecture is then depicted before a presentation of some performances.

1 Contexte

Depuis quelques années déjà, le marché de la télévision numérique ne cesse de s'accroître. La baisse des prix des démodulateurs et l'émergence de nouveaux standards en sont les principales causes. Aujourd'hui, on peut citer bon nombre de standards existants se partageant le marché mondial, tels que DVB en Europe, ATSC et FLO en Amérique du Nord, ou encore ISDB-T, DMB-T, T-DMB en Asie. La multiplicité de ces standards est un point critique à la mobilité des récepteurs et l'idée de regrouper plusieurs standards au sein d'un même démodulateur *reconfigurable* peut être envisagée comme une solution à ce problème.

Afin d'obtenir un récepteur multistandards, deux possibilités sont envisageables. La première consistant à implanter en parallèle plusieurs circuits, chacun étant dédié pour tel ou tel standard, permet un développement rapide du récepteur au prix d'un gaspillage en terme de ressources. La seconde consiste en une réutilisation des ressources nécessaires à la démodulation d'un standard en reconfigurant celles-ci pour démoduler un nouveau standard. La quantité de ressources ainsi nécessaires devient d'autant plus intéressante avec l'accroissement du nombre de standards mis en jeu.

Bien que la plupart des standards de diffusion de la télévision numérique exploitent les techniques de modulation multiporteuses Orthogonal Frequency Division Multiplexing (OFDM), ils présentent à l'émission des caractéristiques variées (intervalle de garde, porteuses pilote, codage de canal...). Ceci influence fortement les techniques de démodulation utilisées en réception. La figure 1 présente un schéma bloc des opérations

effectuées par un récepteur OFDM standard.

Lors de nos travaux, nous nous sommes focalisés sur les standards européen (DVB-T) [1], japonais (ISDB-T) [2] et chinois (DMB) [3] représentant à eux trois une part largement majoritaire du marché mondial. L'opération de TFR étant des plus imposantes dans ce type de récepteur, nous nous intéressons dans cet article à l'implantation d'une TFR *reconfigurable* pouvant travailler sur un nombre de 2048 (2K), 4096 (4K), 8192 (8K) ou 3780 points, tailles utilisées par les standards étudiés.

Malgré une riche littérature sur les architectures reconfigurables pour la TFR [7] [8] [9], celles-ci n'exploitent que des tailles puissances de 2. Nos travaux portent alors sur l'intégration d'une possibilité supplémentaire permettant de traiter une taille de 3780 points. Bien que restreinte à ces tailles de TFR, l'architecture proposée peut-être envisagée dans bien d'autres cas de figures.

La section qui suit présente donc les différents algorithmes de TFR utilisés. Nous proposons ensuite l'architecture matérielle développée permettant l'utilisation de l'un ou l'autre de ces algorithmes avant d'en étudier les performances puis de conclure sur les perspectives à venir.

2 Les algorithmes de TFR

2.1 Radix et Mixed-Radix :

La transformée de Fourier discrète (TFD) sur N points a pour expression : $X(k) = \sum_{n=0}^{N-1} x(n) \times e^{-j2\pi \frac{nk}{N}}$. Effectuer une telle opération sans aucune forme d'optimisation nécessite alors l'intervention de N^2 multiplications complexes. En 1965,

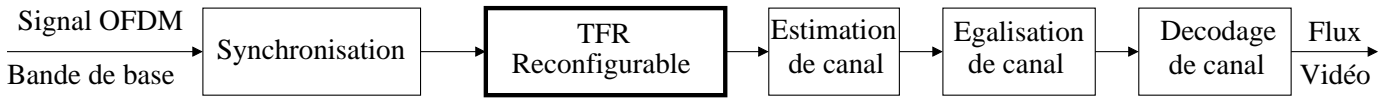


FIGURE 1 – Schéma bloc générique d'un démodulateur OFDM

Cooley et Tukey présentent l'algorithme *Radix-2* [4] réduisant la complexité à l'ordre $N \log_2(N)$ multiplications complexes. Cet algorithme applicable pour des tailles de TFR exprimées en 2^m , $m \in \mathbb{N}$ s'exécute sur m étages faisant intervenir chacun 2^{m-1} papillons *Radix-2*. En sortie du $i^{\text{ème}}$ étage, $\frac{2^m}{2^i}$ TFR de taille 2^i ont été calculées. On obtient ainsi en sortie du dernier étage une TFR de la taille désiré de $2^m = N$ points.

Bien que les papillons *Radix-2* ne fassent intervenir que des multiplications triviales par $+/-1$, le calcul de la TFR de taille 2^m fait intervenir des multiplications complexes entre les différents étages par des termes W_N^x appelés *twiddle factors*.

Cet algorithme, valable en base 2, peut en fait être étendu pour toute taille $N = \prod N_i$ donnant une complexité résultante de l'ordre de $N \sum N_i$. Il est appelé *Mixed-Radix*. Un grand avantage de la famille des algorithmes *Radix* réside dans la régularité du cheminement (*mapping*) des données.

2.2 L'algorithme de Winograd :

En 1978, Winograd fournit le *Winograd Fourier Transform Algorithm* (WFTA) [5], permettant de réduire le nombre de multiplications complexes à l'ordre N au prix de quelques additions supplémentaires. Cette réduction de la complexité intervient sur deux niveaux.

D'un côté la complexité de calcul des TFD élémentaires est réduite de l'ordre N^2 à l'ordre N . Ce point représente une optimisation pour une décomposition de la TFD faisant intervenir des facteurs différents de 2 ou 4 (facteurs pour lesquels les papillons de l'algorithme *Radix* présentent les mêmes avantages).

La seconde optimisation, obtenue par un cheminement astucieux du flot de données, permet la suppression des *twiddle factors* nécessaires via un algorithme de type *Radix*. Cette seconde optimisation n'est cependant valable que si la décomposition de la taille $N = \prod N_i$ fait intervenir des termes N_i premiers entre eux. Le nouveau cheminement des données présente cependant une moins bonne régularité, le complexifiant d'avantage.

3 Architecture de TFR reconfigurable proposée

La TFR peut se décomposer en blocs de type WFTA ou Radix. Par exemple, la TFR sur $N = 64$ peut se décomposer en $N = 4 \times 4 \times 4$ mettant en oeuvre l'algorithme du Radix-4. Dans le cas de la TFR sur $N = 3780$ points, elle peut se décomposer en $N = 7 \times 5 \times 4 \times 3 \times 3 \times 3$, ce qui nous amène à utiliser

les algorithmes WFTA7, WFTA5, WFTA4 et WFTA3. Cette décomposition permet ainsi d'obtenir une architecture reconfigurable WFTA/Radix basée sur une structure pipelinée permettant le calcul des TFR 2K, 4K, 8K ou 3780 points comme présentée sur la figure 2. Le tableau 1 présente quant à lui, les différentes configurations possibles des modules selon la taille de TFR souhaitée.

3.1 Description de l'architecture

L'architecture proposée se décompose en six étages. Chaque étage fait intervenir un module de traitement WFTA/Radix auquel sont connectées deux mémoires de type Random Acces Memory (RAM) servant à stocker les données d'entrée et de sortie du module. Afin de pouvoir traiter des opérations sur des paires d'échantillons sans ajouter de latence supplémentaire, ces mémoires sont de type double ports. Ces mémoires sont dimensionnées pour stocker un symbole OFDM complet en taille 8K. Ainsi la taille de ces block RAM est de $2 \times 16 \times 8192 = 262144$ bits

En sortie de chaque module les données sont, soit directement stockées en mémoire dans le cas de l'algorithme WFTA, soit dirigées vers un block multiplieur complexe permettant d'appliquer les *twiddle factors* dans le cas de l'algorithme Radix. Ces coefficients complexes sont stockés dans les blocs RAM *Twiddle Factors*. Les buffers d'entrée et de sortie servent quant à eux à stocker respectivement les échantillons temporels et fréquentiels.

Les données d'entrée sont représentées sur 32 bits (16 bits pour la partie réelle et 16 bits pour la partie imaginaire). De la même manière les *twiddle factors* sont aussi représentés sur 32 bits.

3.2 Description d'un module

Chaque module se décompose en trois étages. Un premier étage effectue des additions sur les échantillons d'entrée. Les résultats de ces opérations sont multipliés par un coefficient au second étage puis stockés dans des registres. Enfin un dernier étage effectue les additions de sortie dont les résultats sont présentés vers l'extérieur du module.

3.3 Principe de fonctionnement

Les échantillons formant le symbole OFDM S_n entrant dans la structure sont stockés dans le buffer d'entrée. Pendant ce temps le Module 1 exécute les opérations du premier étage de la TFR sur le symbole OFDM précédant S_{n-1} . Plus généralement,

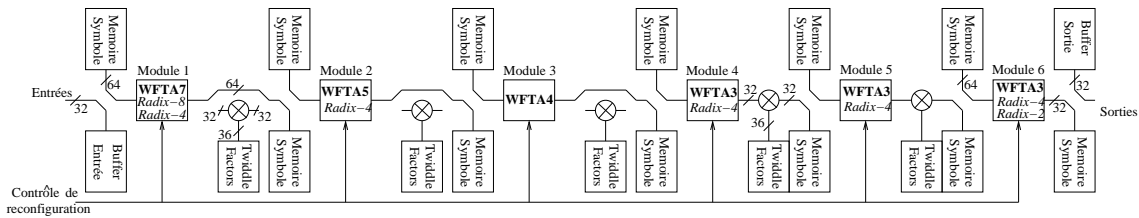


FIGURE 2 – Architecture d’une TFR reconfigurable 2K, 4K, 8K ou 3780 points

Taille de TFR	Configuration Module 1	Configuration Module 2	Configuration Module 3	Configuration Module 4	Configuration Module 5	Configuration Module 6
3780	WFTA7	WFTA5	Radix-4	WFTA3	WFTA3	WFTA3
2048	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4	Radix-2
4096	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4
8192	Radix-8	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4

TABLE 1 – Modes de reconfiguration de la structure

chaque module i exécute les opérations de l’étage i de la TFR sur le symbole S_{n-i} .

Une fois le symbole S_n stocké en mémoire, la mémoire d’entrée de l’étage i est commutée et devient mémoire de sortie de l’étage $i-1$. De la même manière la mémoire de sortie de l’étage i , contenant le symbole OFDM S_{n-i} après traitement par l’étage i , devient mémoire d’entrée de l’étage $i+1$. Les buffers d’entrée et de sortie deviennent respectivement mémoire d’entrée du premier étage et mémoire de sortie du dernier étage et réciproquement.

Les traitements effectués au sein d’un module se font selon une structure pipelinée alors que tous les modules agissent de façon parallèle. La multiplication par un twiddle factor se fait, si nécessaire, juste avant le stockage du résultat en mémoire, en ajoutant cette étape de traitement dans le pipeline du module associé.

3.4 Reconfiguration de l’architecture

La reconfiguration de l’architecture passe par trois points. Le premier est la reconfiguration des modules de traitement en WFTA ou en Radix. Le second consiste en la configuration du cheminement des données à travers les multiplieurs twiddle factor. Le troisième concerne la présence en mémoire des twiddle factors appropriés à la taille de TFR sélectionnée. En ce qui concerne les deux premiers points, la reconfiguration se fait par un contrôle adéquat des différents registres, multiplexeurs, mémoires, additionneurs et multiplieurs de l’architecture. Ceci n’implique que peu de temps de reconfiguration par le chargement d’un nouveau flux de contrôle de la structure. En ce qui concerne le dernier point, la présence en mémoire des différents twiddle factors est alors nécessaire pour ne pas introduire de temps de chargement de ces derniers.

4 Performances de l’architecture

L’architecture proposée a été implanté sur cible FPGA Stratic III EP3SE50F484C2. Ce circuit est en fait employé uniquement à des fins de prototypage de l’architecture. Un ASIC est envisagé comme cible finale permettant ainsi une plus grande densité d’intégration ainsi que des performances, notamment en termes de fréquence de fonctionnement et de consommation, plus importantes.

4.1 Ressources utilisées

Le tableau 2 présente une comparaison des ressources utilisées par l’architecture décrite avec celles nécessaires au fonctionnement des IP proposées par Altera pour réaliser une TFR 2K, 4K ou 8K. On peut alors observer que l’architecture proposée nécessite moins de ressources logiques que les IPs d’Altera bien que celles-ci ne puissent effectuer de TFR que pour des tailles puissances de 2. On remarque également que les ressources mémoires sont les plus consommatrices. Ceci est due à une utilisation des mémoires en *ping-pong* permettant d’atteindre d’importants débits.

4.2 Performances temporelles

Le prototypage sur plateforme FPGA permet d’atteindre une fréquence d’horloge de 100MHz. Le tableau 3 présente les latences des modules en nombre de cycles d’horloge selon la taille de la TFR considérée. Il apparait que les configurations WFTA7 et Radix-8 présentent les latences les plus importantes, relativement au nombre de points de la TFR et vont donc fixer les limites en termes de débits symbole OFDM de la structure entière.

Dans un contexte de diffusion, comme celui de la TV numérique, les contraintes en termes de débit ne sont pas flexibles.

	Cellules logiques	Registres	Bits mémoires	Blocs DSP
Disponibilité Stratix III	38,000	38,000	5,455,872	384
Architecture proposée	4,077	2,544	3,769,080	44
Altera 2K IP	4,138	6,943	208,329	40
Altera 4K IP	4,557	7,530	425,563	40
Altera 8K IP	5,270	8,670	884,785	48

TABLE 2 – Ressources utilisées sur le Stratix III EP3SE50F484C2

Les opérations de TFR doivent alors être effectuées dans un temps majoré par la durée d'un symbole OFDM avec son intervalle de garde. Le tableau 4 recense cette durée pour les standards considérés selon la taille de la TFR employée et pour la taille minimale d'intervalle de garde ainsi que l'équivalent en nombre de cycle d'horloge à 100MHz.

Taille de TFR	3780	2K	4K	8K
WFTA7	11351	-	-	-
WFTA5	7570	-	-	-
WFTA3	3787	-	-	-
Radix-8	-	-	-	16398
Radix-4	3787	2055	4103	8199
Radix-2	-	2055	-	-

TABLE 3 – Temps d'exécution des différents modules en nombre de cycles d'horloge selon la configuration

	DVB	ISDB	DMB
3780	-	-	555 μ s
	-	-	55500
2048	231 μ s	259 μ s	-
	23100	25900	-
4096	462 μ s	519 μ s	-
	46200	51900	-
8192	924 μ s	1039 μ s	-
	82400	103900	-

TABLE 4 – Durée minimum d'un symbole OFDM et son équivalent en nombre de cycles d'horloge à 100MHz

Il apparait que les contraintes de débits sont largement respectées. Ceci permet donc d'optimiser l'architecture proposée en termes d'utilisation des ressources mémoires, par l'exploitation d'une structure réduite à un seul module de calcul reconfigurable exécutant tous les étages de la TFR.

5 Conclusion

Cette nouvelle architecture permet d'effectuer l'opération de Transformée de Fourier Rapide selon différents algorithmes contrairement aux structures classiques où la reconfiguration, réalisable uniquement en Radix, consiste en la suppression d'un certain nombre d'étages de la chaîne. Les contraintes tempo-

relles des différents standards sont respectées et la marge obtenue nous permet d'envisager une optimisation principalement en termes d'occupation des blocs de mémoire.

Une étude sur les impacts de la quantification sur le bruit généré est pour l'heure à l'étude. Cette étude permettra ainsi une réduction supplémentaire de la consommation de bits mémoires.

Bien que non étudié dans cet article, l'architecture proposée peut aisément être étendue pour une utilisation de celle-ci en modes 16K et 32K, modes ayant été ajouté dans les spécifications de la nouvelle norme de diffusion de la télévision numérique européenne DVB-T2. Cependant une augmentation conséquente en terme de ressources mémoire est alors à prévoir.

Bien qu'implantée sur FPGA, l'architecture proposée est destinée à une cible de type ASIC offrant de meilleurs performances, notamment en termes de fréquence d'horloge.

Remerciements

Ce travail a été supporté par le projet Mobile TV World, labélisé par le pôle Images et Réseaux et financé par la région Bretagne.

Références

- [1] ETSI EN 300 744 v1.5.1. *DVB-T; Framming structure, channel coding and modulation for DTV*. ETSI 2004.
- [2] ARIB STD-B31 v1.6. *Transmission system for DTTB*. ARIB 2001.
- [3] China *Chinese standard for DTTB*. 2006.
- [4] J.W. Cooley et J.W. Tukey *An algorithm for the machine calculation of complex fourier series*. math. Comp., v.19, 1965, pp297-301.
- [5] S. Winograd *On computing the discrete fourier transform*. math. Comp., v.32, 1978, pp175-199.
- [6] H.F. Silverman *An introduction to programming the WFTA*. IEEE Trans., v.ASSP-25, 1977, pp152-164.
- [7] J.-C. Kuo, C.-H. WEN, C.-H. Lin and A.-Y. Wu, *VLSI design of a variable length FFT/IFFT processor for OFDM-based communication systems*. EURASIP journal on Applied signal processing, 2003, pp1306–1316.
- [8] Altera, *FFT mega core function user guide*, March 2009
- [9] Xilinx, *Fast Fourier Transform v6.0*, September 2008