

Projet de Fin d'Etude

Traitement du signal et communication numérique sur
FPGA

*Spécialité Electronique et Signal pour les Systèmes
Embarqués*

ENSIETA (Ecole Nationale Supérieure d'Ingénieurs)

Kermarrec Florent

26 août 2008

THALES



Maître de stage : Frédéric SAVINA

Responsable école : Luc JAULIN

1 Remerciements

Je tiens tout d'abord à remercier Frédéric SAVINA pour son implication dans son rôle de maître de stage, son aide et sa sympathie tout au long de cette année de professionnalisation.

Je tiens aussi à remercier particulièrement Jacques LE GUEN et Laurent TREGUER pour leur nombreux conseils et toute l'aide qu'ils ont pu m'apporter au cours de ce stage.

Je remercie également toute l'équipe du service architecture numérique pour son accueil et sa sympathie;-).

1	Remerciements	2
2	Introduction	5
2.1	Contexte du stage	5
2.2	Finalité du PFE	5
3	Analyse spectrale voie courte/voie longue sur FPGA	8
3.1	Présentation du projet d'analyse spectrale	8
3.1.1	Fenêtrage	8
3.1.2	Overlapping	9
3.1.3	Algorithme de transformation T/F retenu	9
3.2	Architecture retenue	11
3.2.1	Justification	11
3.2.2	Fenêtrage	12
3.2.3	Mémoire tampon pour la FFT	13
3.2.4	FFT	14
3.2.5	Mémoire tampon de reconstitution	15
3.2.6	Reconstitution	16
3.3	Architecture de la voie courte d'analyse N1 points	19
3.4	Architecture de la voie longue d'analyse N2 points	20
3.5	Evaluation du module sous Matlab et simulation modelsim	21
3.5.1	Test de la voie courte N1 points	21
3.5.2	Test de la voie courte N2 points	23
3.5.3	Etude de la répétitivité et de la cohérence des différentes voies	24
3.5.4	Etude de la répétitivité	25
3.6	Conclusion des travaux sur l'analyse spectrale	26
4	Pile UDP/IP Gigabit Ethernet sur FPGA	28
4.1	Mise en situation	28
4.1.1	Présentation	28
4.1.2	Performances attendues	29
4.2	Architecture du module	30
4.3	Validation	32
4.4	Ajout de la connectivité Gigabit Ethernet à une carte de traitement numérique	33
5	Conclusions et perspectives du PFE	35
A	Présentation de Thales et du service architecture numérique	36
	Références	37

Table des figures

1	Exemple d'utilisation pour un récepteur numérique	6
2	Vue d'une carte à coprocesseurs FPGAs délocalisés	7
3	Effets du fenêtrage sur le signal	8
4	Overlapping	9
5	FFT avec séparation des $N/2$ données	10
6	FFT avec séparation des données paires/impaires	10
7	Démultiplexeur 4 voies	11
8	Ordonnancement du traitement des données	11
9	Synoptique du module de fenêtrage	12
10	Mémoire tampon pour la FFT	13
11	FFT	14
12	FFT	15
13	FFT	16
14	Reconstitution réelle	17
15	Reconstitution imaginaire	18
16	Bloc FFT N1	19
17	Overlap 0 de la voie d'analyse N1 points	20
18	Signal à transposer dans le domaine fréquentiel	21
19	Module de la FFT N1 points calculée sous Matlab	22
20	Module de la FFT N1 points simulée sous modelsim	22
21	Erreur absolue entre le calcul et le modèle simulé	23
22	Module de la FFT N2 points calculée sous Matlab	23
23	Module de la FFT N2 points simulée sous modelsim	24
24	Erreur absolue entre le calcul et le modèle simulé	24
25	Cohérence de la voie N1 points	25
26	Cohérence de la voie N2 points	25
27	Test de répétitivité voie N1 points	26
28	Test de répétitivité voie N2 points	26
29	Situation de la pile UDP/IP dans le système	28
30	Encapsulation des données dans le cas d'un streaming	29
31	Architecture de la pile UDP/IP	30
32	Architecture de la carte de traitement numérique	33
33	Routage des packets et partage de l'accès	34

2 Introduction

2.1 Contexte du stage

L'apparition des premiers systèmes de guerre électronique[5] embarqués date de la seconde guerre mondiale. La technologie de l'époque et la faible compacité des dispositifs ne permettaient pas d'embarquer des traitements complexes, les tous premiers systèmes ne servaient ainsi qu'à leurrer la station radar au sol, masquant la présence de l'appareil et lui donnant dès lors l'occasion d'échapper à la surveillance de l'ennemi. Le contexte de la guerre froide n'allait qu'accélérer la mise en place de tels systèmes à bord des avions militaires, si bien que de nos jours, ceux-ci font appel aux toutes dernières technologies émergentes. La technologie analogique bridant auparavant l'imagination des chercheurs, l'ère du tout numérique offre de nouvelles voies de recherches et d'expérimentations, donnant naissance à des systèmes de plus en plus riches en fonctionnalités mais aussi de plus en plus complexes à concevoir. L'arrivée de composants logiques reconfigurables (FPGA¹) ces dernières années permet d'embarquer des systèmes de plus en plus compacts, puissants et surtout polyvalents.

Jusqu' alors cantonnés aux applications purement numériques, les FPGAs gagnent en polyvalence, par l'adjonction de fonctions spécialisées à l'intérieur même de la puce, mais aussi par leur compacité croissante d'années en années. De nombreuses opportunités s'ouvrent alors aux ingénieurs, comme le traitement du signal appliqué à la guerre électronique, ou encore des communications numériques ultra-rapides entre les modules d'un système de guerre électronique.

Dans le cadre de l'étude d'un récepteur de guerre électronique ELINT², Thales Systèmes Aéroportés met en oeuvre ces nouvelles technologies, afin de détecter et d'analyser des signaux tels que les émissions radars ou les émissions électromagnétiques de divers appareils électroniques, et ainsi connaître au mieux la situation tactique. Les systèmes effectuent des balayages de larges bandes du spectre électromagnétique, ce qui constitue une détection préliminaire pour les interceptions COMINT³. Les unités ELINT, sont polyvalentes, et sont généralement chargées de l'interception et de l'analyse dans les domaines SIGINT⁴ et COMINT. Elles sont également chargées d'évaluer, et de mettre en oeuvre les meilleurs moyens de guerre électronique, dans le but de brouiller, tromper ou détruire les émetteurs et récepteurs électroniques adverses.

Ces nouvelles technologies sont également utilisées dans les systèmes ESM⁵, il s'agit ici d'effectuer une mesure des émissions radioélectriques de l'environnement lors d'une opération, et d'identifier les menaces à partir des signaux reçus et d'une base de données.

2.2 Finalité du PFE

Un récepteur numérique se doit ainsi de traiter un maximum d'informations en un laps de temps très restreint, permettant la mise en place des moyens de guerre électronique adaptés à la menace. Lors du balayage des bandes du spectre électromagnétique, les principales transformations appliquées au signal sont des FFTs réalisant la transposition T/F⁶. Il faut de plus maximiser le débit d'information entre les systèmes FrontEnd et BackEnd pour permettre une meilleure prise de décision tactique en réalisant, par exemple, des traitements plus poussés de l'information après première détection par le système FrontEnd qui permettront de mieux connaître

¹FPGA : Field-Programmable Gate Array [4]

²ELINT : Electronic Signals INTelligence

³COMINT : COMmunication INTelligence

⁴SIGINT : SIGnals INTelligence

⁵ESM : Electronic Support Measures

⁶T/F : Temps/Fréquence

et identifier la menace par la suite, et ainsi mieux s'y préparer.

Mon projet de fin d'étude se décompose en 2 parties :

- L'étude de la faisabilité d'une solution d'analyse spectrale disposant d'une voie d'analyse courte et d'une voie d'analyse longue sur FPGA dans le cadre d'un récepteur numérique.
- L'étude de l'architecture d'une pile UDP/IP hardware sur FPGA pour une connectivité réseau Gigabit Ethernet entre un système FrontEnd et un système BackEnd de guerre électronique.

La combinaison de ces études permettra par exemple l'amélioration d'un système de guerre électronique ELINT (si ces solutions technologiques sont retenues), jusqu'alors bien plus limité en résolution de transposition T/F et en débit de transmission vers le système BackEnd.

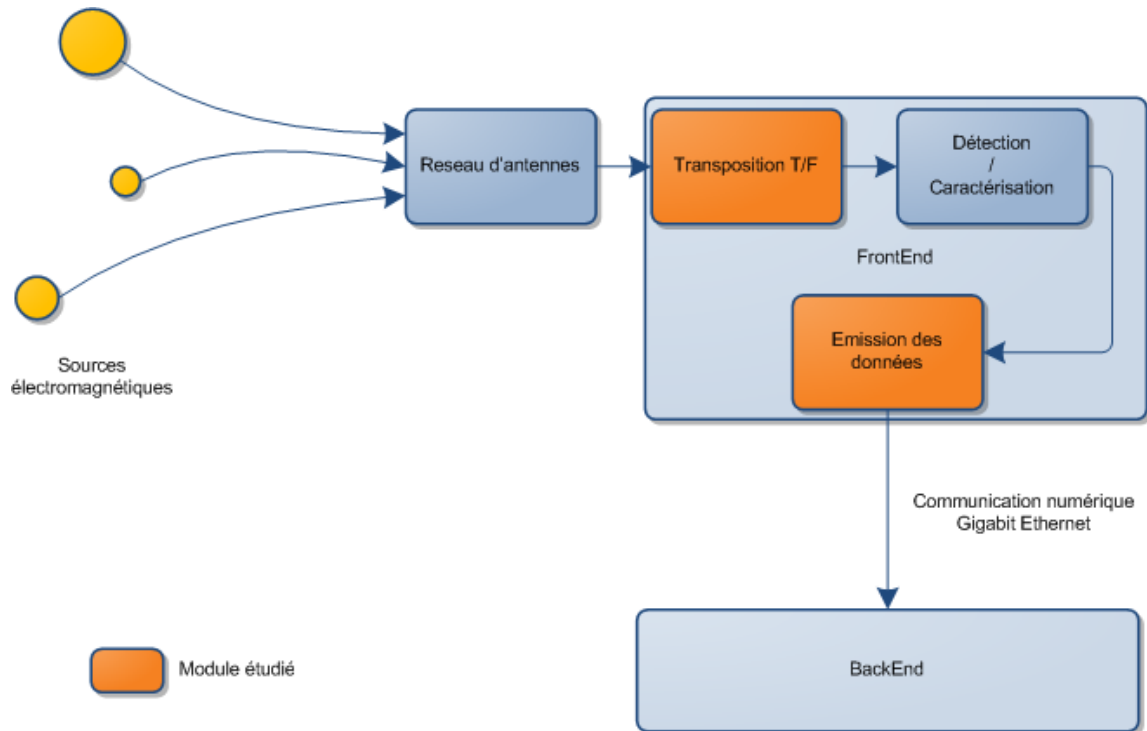


FIG. 1 – Exemple d'utilisation pour un récepteur numérique

L'analyse spectrale sera présentée en première partie de rapport.

La pile UDP/IP est quant à elle déjà utilisée dans l'amélioration d'une carte de traitement numérique. Pour certains traitements numériques en guerre électronique les FPGAs sont utilisés en tant que coprocesseurs délocalisés car parfaitement appropriés à ce type de calcul. Il peut en effet exister un rapport de plus de 100 entre le temps d'exécution d'un calcul sur une architecture polyvalente (à base de microprocesseur d'ordinateur par exemple) et celui obtenu sur une architecture spécialisée (ici un FPGA).

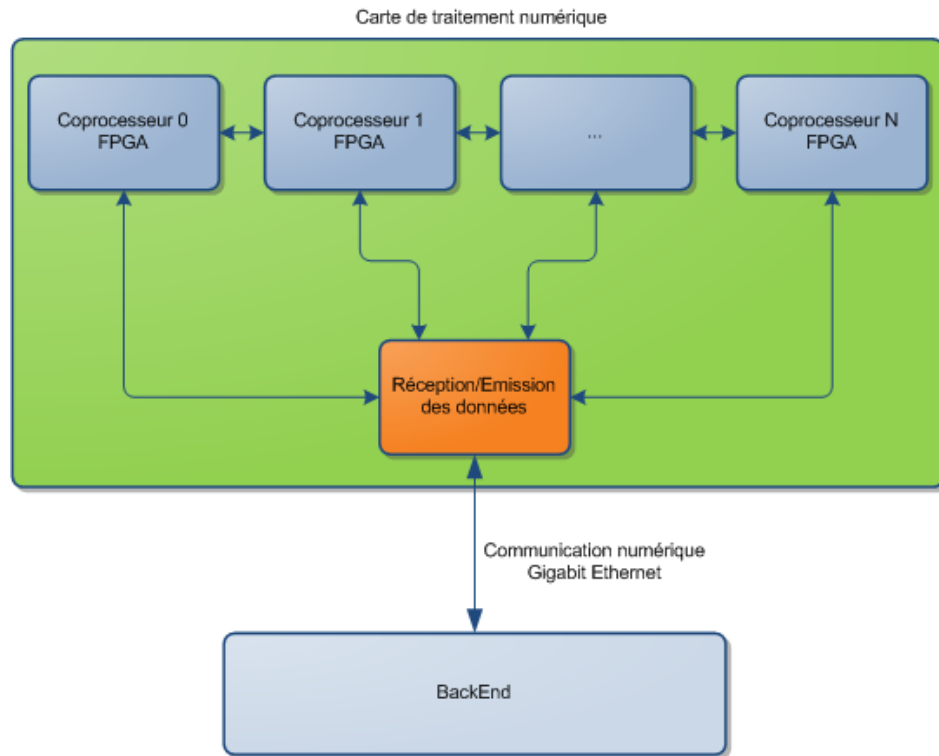


FIG. 2 – Vue d'une carte à coprocesseurs FPGAs délocalisés

La pile UDP/IP sera présentée dans une seconde partie de rapport, ainsi que son utilisation dans une carte de traitement à coprocesseurs FPGA.

Une partie du PFE s'inscrit donc dans le cadre d'études amonts menées au sein du service d'architecture numérique du centre Thales Systèmes Aéroportés (TAS) de Brest, afin de rendre compte des possibilités de futurs systèmes de guerre électronique. Une autre partie s'attache plus au processus de développement d'une amélioration de carte. **Du fait de la confidentialité des résultats obtenus, je ne serai pas en mesure de présenter certains d'entre-eux, notamment des choix architecturaux ou certaines performances obtenues.**

3 Analyse spectrale voie courte/voie longue sur FPGA

3.1 Présentation du projet d'analyse spectrale

Le but d'une chaîne d'analyse spectrale temps réel est d'effectuer une transposition du domaine temporel au domaine fréquentiel. La problématique principale qui se pose aujourd'hui en architecture numérique [3] est la capacité à traiter la totalité des échantillons en entrée, donc de faire fonctionner le module à la vitesse d'acquisition des échantillons. Les contraintes sont dès lors très fortes sur l'architecture du module, car bien souvent, les fréquences d'échantillonnages sont intenable par les composants actuels, il est donc nécessaire de paralléliser les traitements de façon très forte.

Abordons tout d'abord les contraintes inhérentes à une transposition T/F sur des échantillons discrets.

3.1.1 Fenêtrage

D'après la théorie, une transformée de Fourier doit se calculer sur un nombre entier de périodes. Dans le cas contraire des raies parasites apparaissent dans le spectre fréquentiel. Notre module réalise une FFT⁷ du signal temporel, la forme à priori du signal ne peut être connue, l'hypothèse d'un nombre entier de période n'a alors plus aucun sens. Pour limiter l'apparition de ces raies parasites, des fonctions d'apodisation (ou de fenêtrage) sont appliquées au signal temporel. Les bords de fenêtre temporelle se retrouvent écrasés, si bien que les effets de bord disparaissent et n'engendrent plus de raies parasites.

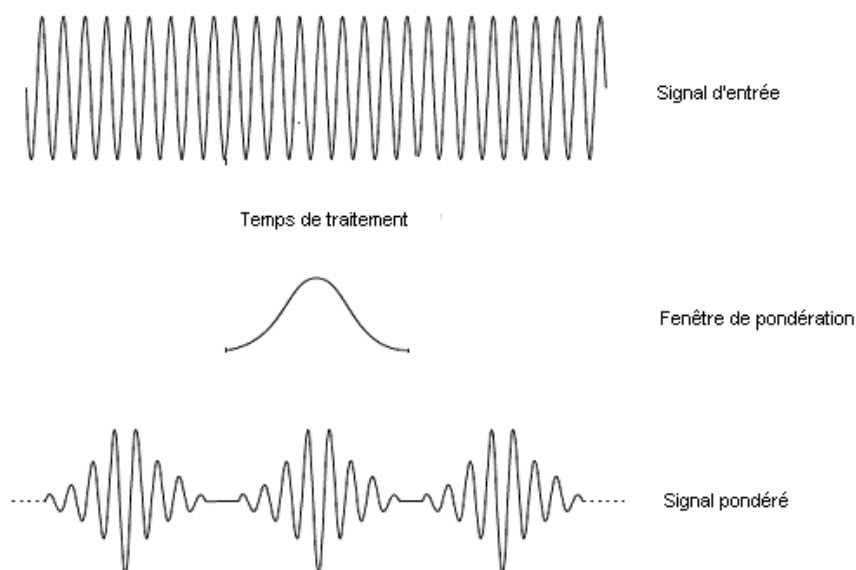


FIG. 3 – Effets du fenêtrage sur le signal

La figure montre une perte d'information dans les bords de fenêtre. La résolution du problème

⁷FFT : Fast Fourier Transform

des raies parasites engendre dès lors un second problème : La perte de l'information dans les zones d'écrasement de la fenêtre ne nous garantit plus la détection de toutes les sources électromagnétiques, il faut donc y remédier.

3.1.2 Overlapping

L'overlapping est une technique qui consiste à traiter les échantillons temporels avec un certain taux de recouvrement. Il est dès lors possible d'éviter la perte d'informations inhérente au fenêtrage en traitant les échantillons temporels sur plusieurs voies parallèles, et en décalant la fenêtre de pondération sur les différentes voies.

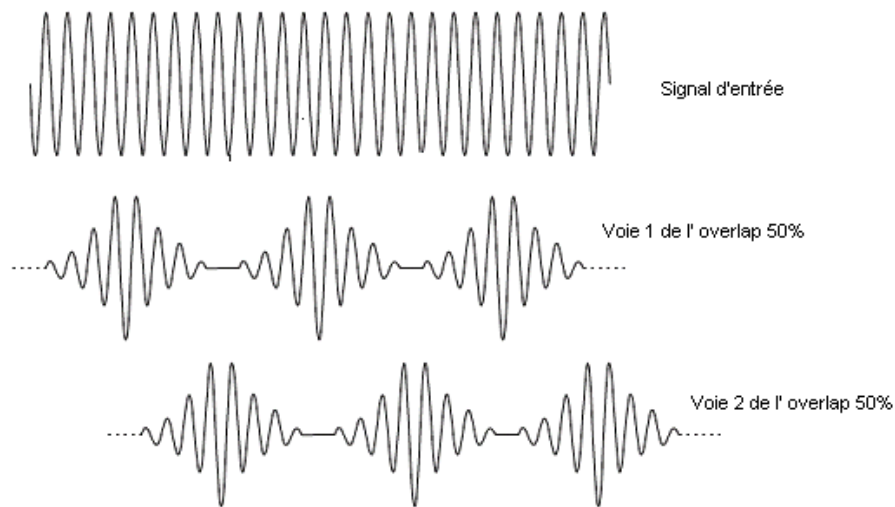


FIG. 4 – Overlapping

Le choix du taux de recouvrement détermine directement le nombre de voies de traitement (ie 50% \rightarrow 2 voies, 66% \rightarrow 3 voies...) mais la détermination de celui-ci dépend de la fenêtre de pondération utilisée et des ressources disponibles sur le composant cible. Dans le cas de notre module, un overlapping de 50% est le juste compromis performance/ressources utilisées.

3.1.3 Algorithme de transformation T/F retenu

Un algorithme de FFT traite de données complexes en entrées, or notre système ne traite que des données réelles. Il est possible de n'utiliser que l'entrée réelle de la FFT, le résultat en sortie serait correct, mais l'algorithme étant optimisé pour des données complexes, nous n'obtiendrons pas les performances optimales.

Différentes méthodes permettent d'optimiser l'utilisation de FFTs complexes pour des données réelles en entrée :

On travaille sur N échantillons, donc sur une FFT N points.

- Placer $N/2$ données sur la partie réelle, les $N/2$ suivantes sur la partie imaginaire et réaliser une opération de reconstitution en aval.

- Placer les $N/2$ données paires sur la partie réelle, les $N/2$ données impaires sur la partie imaginaire et réaliser une opération de reconstitution en aval.

D'autres méthodes existent, mais n'auraient pas convenu à notre application.

La première méthode permet de placer très facilement les données sur les entrées de la FFT, au fur et à mesure de leur arrivée. Elle nécessite par ailleurs une opération de reconstitution des données difficile à mettre en oeuvre sur FPGA. La méthode de séparation des données paires et

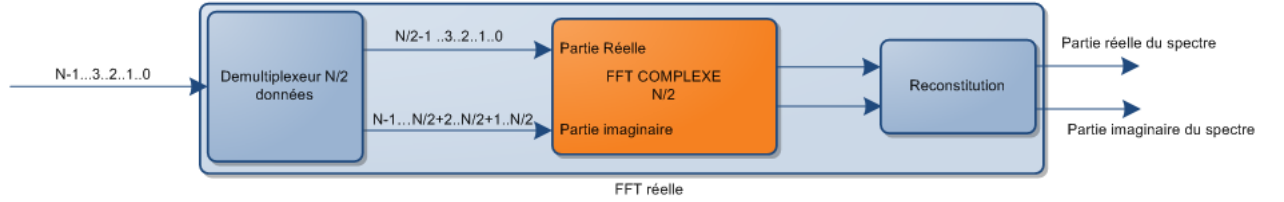


FIG. 5 – FFT avec séparation des $N/2$ données

impaires, possède une opération de reconstitution mieux adaptée à son implémentation. Nous

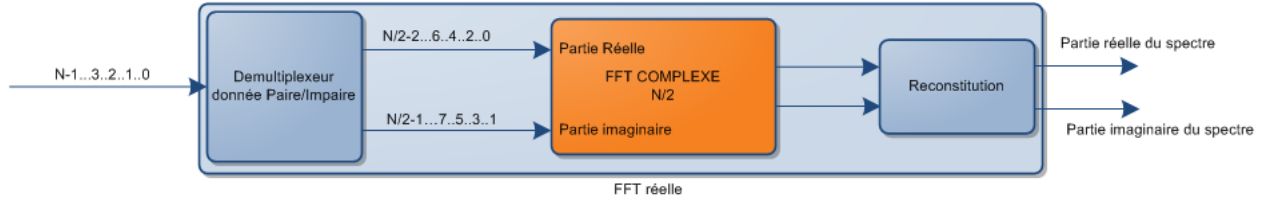


FIG. 6 – FFT avec séparation des données paires/impaires

ne détaillerons pas l'algorithme complet de la première méthode, vite abandonné du fait de sa complexité d'implémentation. Voici l'algorithme retenu pour la reconstitution de la seconde méthode : Notons S le spectre de sortie de la FFT N réelle :

$$Re(S(k)) = \frac{1}{2} \cdot (Re(k) + Re(N - k) + Im(k) \cdot \cos(\frac{\pi \cdot k}{N}) + Im(N - k) \cdot \cos(\frac{\pi \cdot k}{N}) - Re(k) \cdot \sin(\frac{\pi \cdot k}{N}) + Re(N - k) \cdot \sin(\frac{\pi \cdot k}{N}))$$

$$Im(S(k)) = \frac{1}{2} \cdot (Im(k) - Im(N - k) + Re(k) \cdot \cos(\frac{\pi \cdot k}{N}) + Re(N - k) \cdot \cos(\frac{\pi \cdot k}{N}) - Im(k) \cdot \sin(\frac{\pi \cdot k}{N}) - Im(N - k) \cdot \sin(\frac{\pi \cdot k}{N}))$$

Cette méthode permet d'utiliser toutes les ressources de la FFT fournie par le générateur de Xilinx, l'opération de reconstitution se réalise bien sur FPGA.

3.2 Architecture retenue

3.2.1 Justification

Notre module possède deux voies de transposition T/F, une voie courte à N_1 points, et une voie longue à N_2 points. La voie courte offre l'avantage d'une très bonne résolution temporelle tandis que la voie longue apporte elle une meilleure précision fréquentielle. Les 2 voies de traitement sont effectuées en parallèle, le choix du composant cible oriente donc très fortement le choix de la solution implémentée du fait de ses ressources limitées.

Intéressons nous à une seule de ces voies pour mieux comprendre l'architecture du module : Les échantillons de données arrivent à la fréquence F_e , fréquence plus de 3 fois supérieure à la fréquence de fonctionnement des FFTs, 4 voies parallèles de données seront donc nécessaires pour assurer leur traitement en temps réel. Le module devra donc supporter une fréquence de fonctionnement de $\frac{F_e}{4}$.

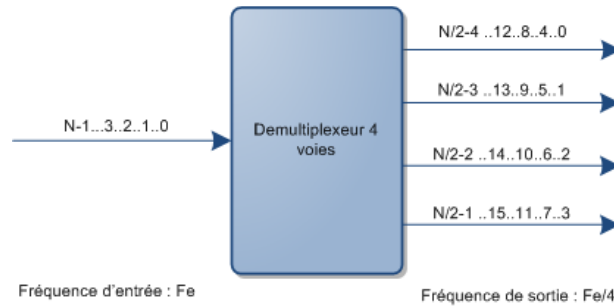


FIG. 7 – Démultiplexeur 4 voies

D'après l'algorithme de traitement, en ne tenant pas compte de l'overlapping, la FFT récupère sur son entrée réelle les échantillons pairs et sur son entrée imaginaire les échantillons impairs. Une simple opération de multiplexage est donc nécessaire, faisant passer la fréquence de fonctionnement du module à $\frac{F_e}{2}$, chose impossible. Il faut donc utiliser 2 FFTs à $\frac{F_e}{4}$ pour le traitement, la première traitant les N premières données, la seconde les N suivantes :

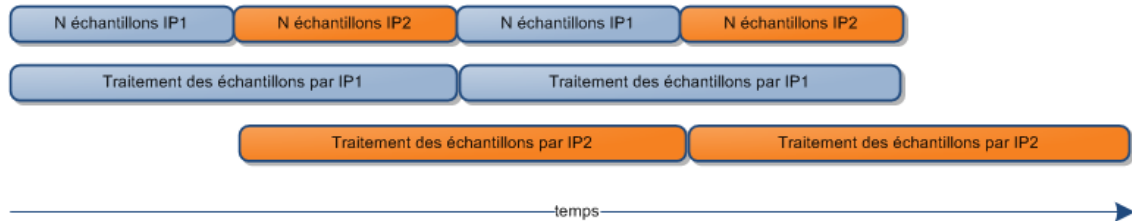


FIG. 8 – Ordonnancement du traitement des données

Ainsi pour chaque voie de l'overlap 2 FFTs sont nécessaires, nous conduisant à 4 FFTs par voie de transposition T/F. Soit pour résumer :

- 4 FFTs complexes $N_1/2$ pour la voie courte N_1 points
- 4 FFTs complexes $N_2/2$ pour la voie longue N_2 points

3.2.2 Fenêtrage

D'après la partie 3.1.1, le fenêtrage doit réaliser une pondération du signal d'entrée pour limiter les effets de bord dans le résultat de la transposition T/F, et ainsi éliminer les raies parasites.

D'après la partie 3.1.3, le module doit également tenir la fréquence de $\frac{F_e}{4}$, avec 4 voies de données en parallèle.

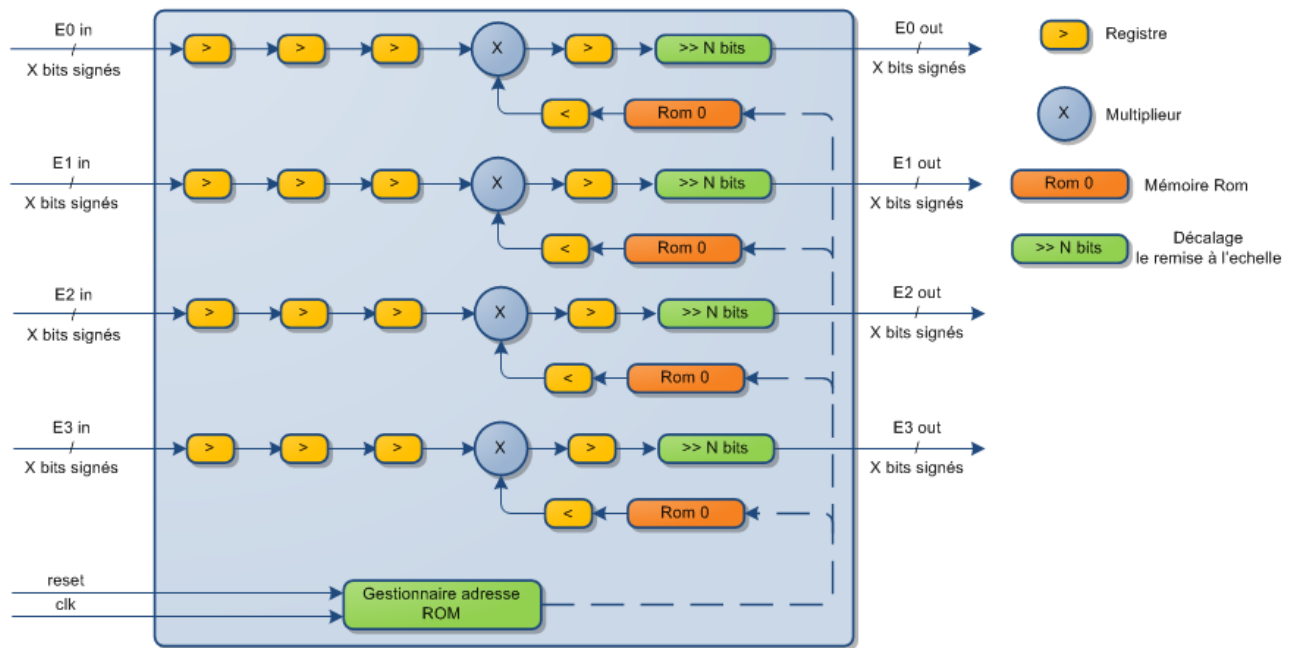


FIG. 9 – Synoptique du module de fenêtrage

Entrées

- E0..3 in : Données des codeurs ADC sur 4 voies
- Init : Init de remise à zéro du pointeur d'adresse des roms
- Clk : Horloge du module

Sorties

- E0..3out : Données de sortie pondérées

Le module fonctionne d'une façon simple, chaque voie réalise le même traitement, à l'aide de coefficients stockés dans les roms, les échantillons sont pondérés. Les données sont registrées en entrées pour garantir la correspondance des données d'entrées avec les données en sortie de rom lors de la multiplication. La sortie de la multiplication est registrée pour optimiser la vitesse de fonctionnement du module, et enfin, une remise à l'échelle des données est réalisée.

La latence du module est de 4 cycles d'horloge.

3.2.3 Mémoire tampon pour la FFT

D'après la partie 3.1.3 l'algorithme de traitement utilise une FFT $N/2$ complexe traitant sur sa voie réelle les échantillons pairs et sur sa voie imaginaire les échantillons impairs. Une réorganisation des données en sortie du module de fenêtrage est donc nécessaire.

La FTT créée à l'aide de l'outil Coregen de Xilinx impose l'ordre des échantillons d'entrée, il faut donc stocker l'équivalent d'une fenêtre de traitement avant de lancer le calcul de la FFT, car l'ordre de prélèvement des données n'est pas connu à priori.

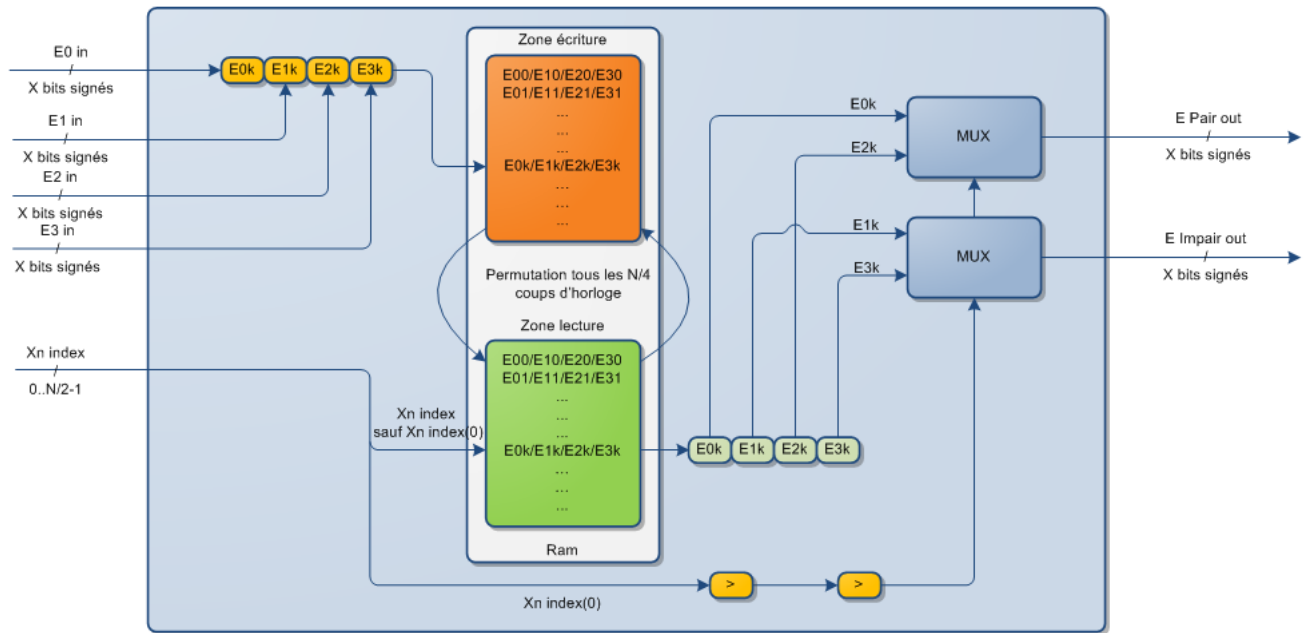


FIG. 10 – Mémoire tampon pour la FFT

Entrées

- E0..3 in : Données en provenance du fenêtrage, sur 4 voies
- xn index : Index de la donnée à sortir (fourni par la FFT)
- Clk : Horloge du module

Sorties

- E pair out : Données paires pour la partie réelle de la FFT
- E impair out : Données impaires pour la partie réelle de la FFT

Les données sont d'abord concaténées pour être stockées en ram. Celle-ci est partagée en 2 zones : une zone de lecture et une zone d'écriture. La lecture dans la ram est 2 fois plus lente que l'écriture dans celle-ci. Le bloc n'enregistrera alors que N premières données sur les 2N données présentées en entrée. Les N autres données étant traitées par une autre mémoire tampon fonctionnant en décalé(cf figure d'ordonnancement). Il y a une permutation des zones d'écriture et de lecture tous les $N/4$ cycles d'horloge. Ainsi à la fin de l'écriture, la ram contenant les N échantillons à traiter

pas en lecture. A la mi-parcours de la séquence de lecture, la ram d'écriture recommence à enregistrer les échantillons.

La FFT Xilinx nous impose de fournir les données en entrée de FFT 3 cycles d'horloge après l'émission de l'index par celle-ci. Il faut donc tenir compte de cette contrainte dans l'architecture. Pour adresser la ram, on utilise le xn index privé du premier bit pour sortir le groupe de 4 données de la ram, puis on fait une sélection des données avec le bit 0 de xn index.

La latence pour l'enregistrement des données est d'1 cycle d'horloge. Pour la lecture des données, celle-ci atteint 3 cycles d'horloge.

3.2.4 FFT

Nous ne nous attarderons pas sur ce module, car simplement généré à l'aide de l'outil Coregen de Xilinx, il est cependant important de connaître son interfaçage avec les autres modules du projet :



FIG. 11 – FFT

Entrées

- xn re : Données réelles d'entrée correspondant au xn index émis 3 coups d'horloge plus tôt
- init : Init de remise à zéro du pointeur d'adresse des roms
- Clk : Horloge du module

Sorties

- xn index : Indice des échantillons à traiter par la FFT
- xk index : Indice des échantillons en sortie de FFT
- xk re : Données réelles du spectre de sortie
- xk im : Données imaginaires du spectre de sortie
- edone : Indique la fin du traitement un cycle d'horloge en avance, permet l'initialisation des blocs suivants.

La latence dépend de N d'une façon non linéaire, elle est en générale supérieure à N .

3.2.5 Mémoire tampon de reconstitution

D'après la partie 3.1.3, l'algorithme de reconstitution fait appel aux échantillons d'indices k ainsi qu'aux échantillons d'indices $N-k$. Il est donc nécessaire de disposer du spectre complet en sortie de FFT complexe pour cette étape.

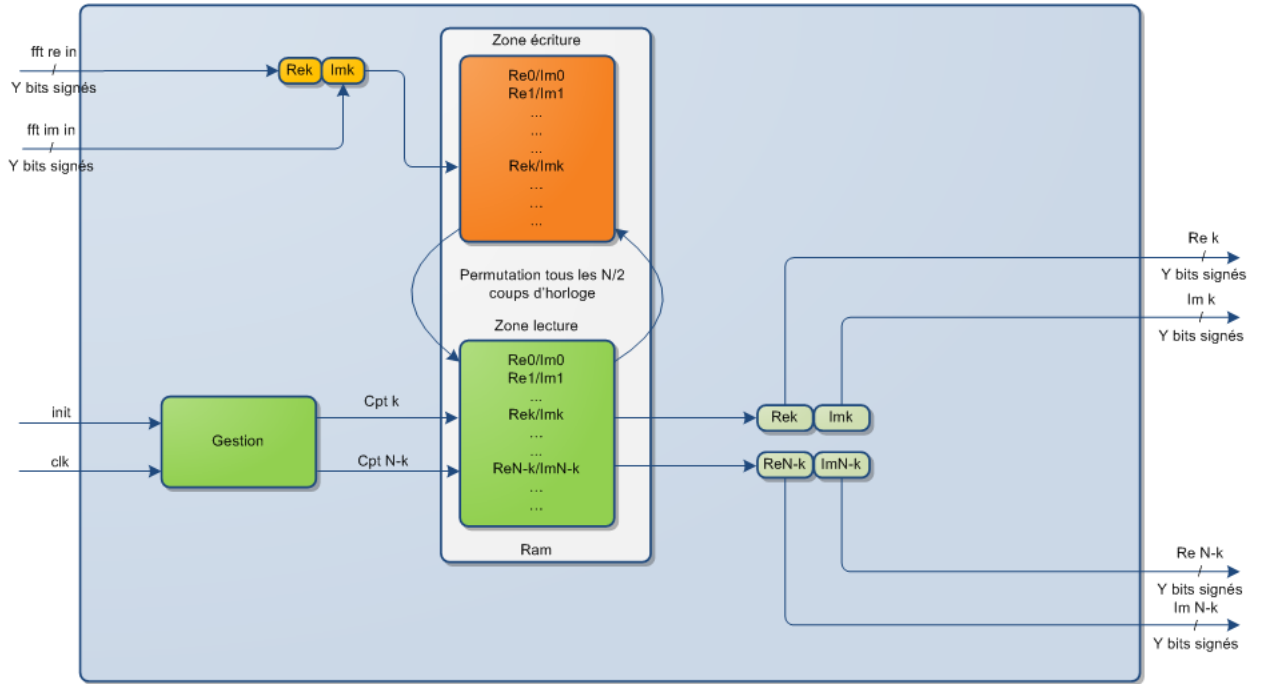


FIG. 12 – FFT

Entrées

- FFT re in : Données réelles d'entrée correspondant au xk index
- FFT im in : Données imaginaires d'entrée correspondant au xk index
- xk index : Index des données d'entrée
- init : Init actif à 1, réinitialise les variables internes du module
- Clk : Horloge du module

Sorties

- $\text{re } k$: Données réelles de sortie d'indice k
- $\text{im } k$: Données imaginaires de sortie d'indice k
- $\text{re } Nk$: Données réelles de sortie d'indice $N-k$
- $\text{im } Nk$: Données imaginaires de sortie d'indice $N-k$

Le bloc réalise une mise en mémoire tampon du spectre de sortie de la FFT suivant les indices fournis par celle-ci, tout en restituant les données du spectre précédent, dans l'ordre nécessaire à la reconstitution. Il est à noter que pour la reconstitution, les indices k et $N-k$ doivent être sortis simultanément, on utilise donc des mémoires doubles ports pour la lecture. L'écriture quant à

elle ne se fait que par un seul port. Il existe quelques subtilités pour une bonne permutation des zones de lecture et d'écriture mais nous ne nous y intéresserons pas dans cette partie descriptive du bloc.

La latence pour l'enregistrement des données est d' 1 cycle d'horloge. Pour la lecture des données, celle-ci atteint 3 cycles d'horloge.

3.2.6 Reconstitution

La reconstitution est l'étape finale de la chaîne de traitement, elle permet de récupérer la FFT des signaux réels de taille N à partir de la FFT complexe de taille N/2. La difficulté principale de ce module est l'utilisation importante d'opérateurs et de fonctions mathématiques. La reconstitution s'effectue en effet suivant la formule : (formule déjà traduite en expressions mathématiques simples, en décomposant la fonction exponentielle en cosinus et sinus)

$$Re(S(k)) = \frac{1}{2} \cdot (Re(k) + Re(N - k) + Im(k) \cdot \cos(\frac{\pi \cdot k}{N}) + Im(N - k) \cdot \cos(\frac{\pi \cdot k}{N}) - Re(k) \cdot \sin(\frac{\pi \cdot k}{N}) + Re(N - k) \cdot \sin(\frac{\pi \cdot k}{N}))$$

$$Im(S(k)) = \frac{1}{2} \cdot (Im(k) - Im(N - k) + Re(k) \cdot \cos(\frac{\pi \cdot k}{N}) + Re(N - k) \cdot \cos(\frac{\pi \cdot k}{N}) - Im(k) \cdot \sin(\frac{\pi \cdot k}{N}) - Im(N - k) \cdot \sin(\frac{\pi \cdot k}{N}))$$

Les fonctions cosinus et sinus seront stockées sous forme de tables dans des roms.

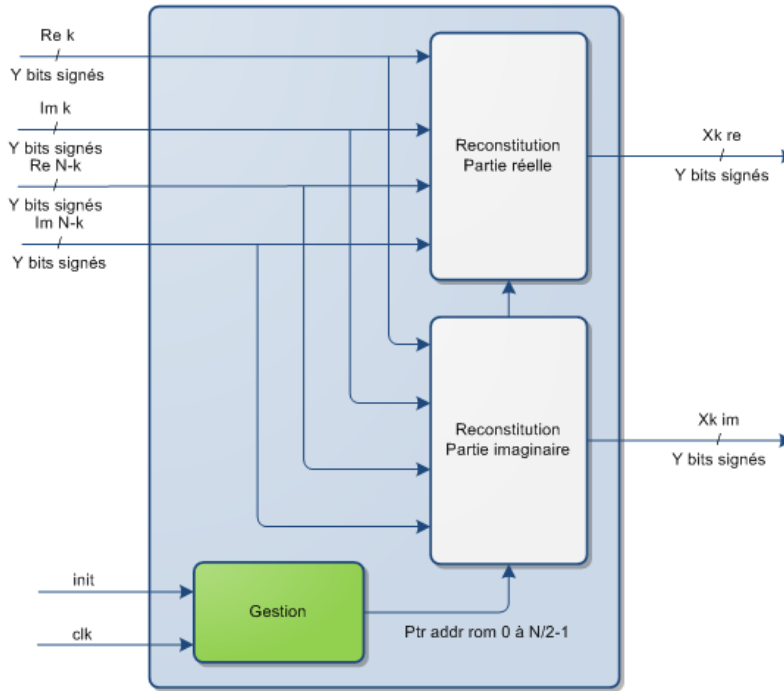


FIG. 13 – FFT

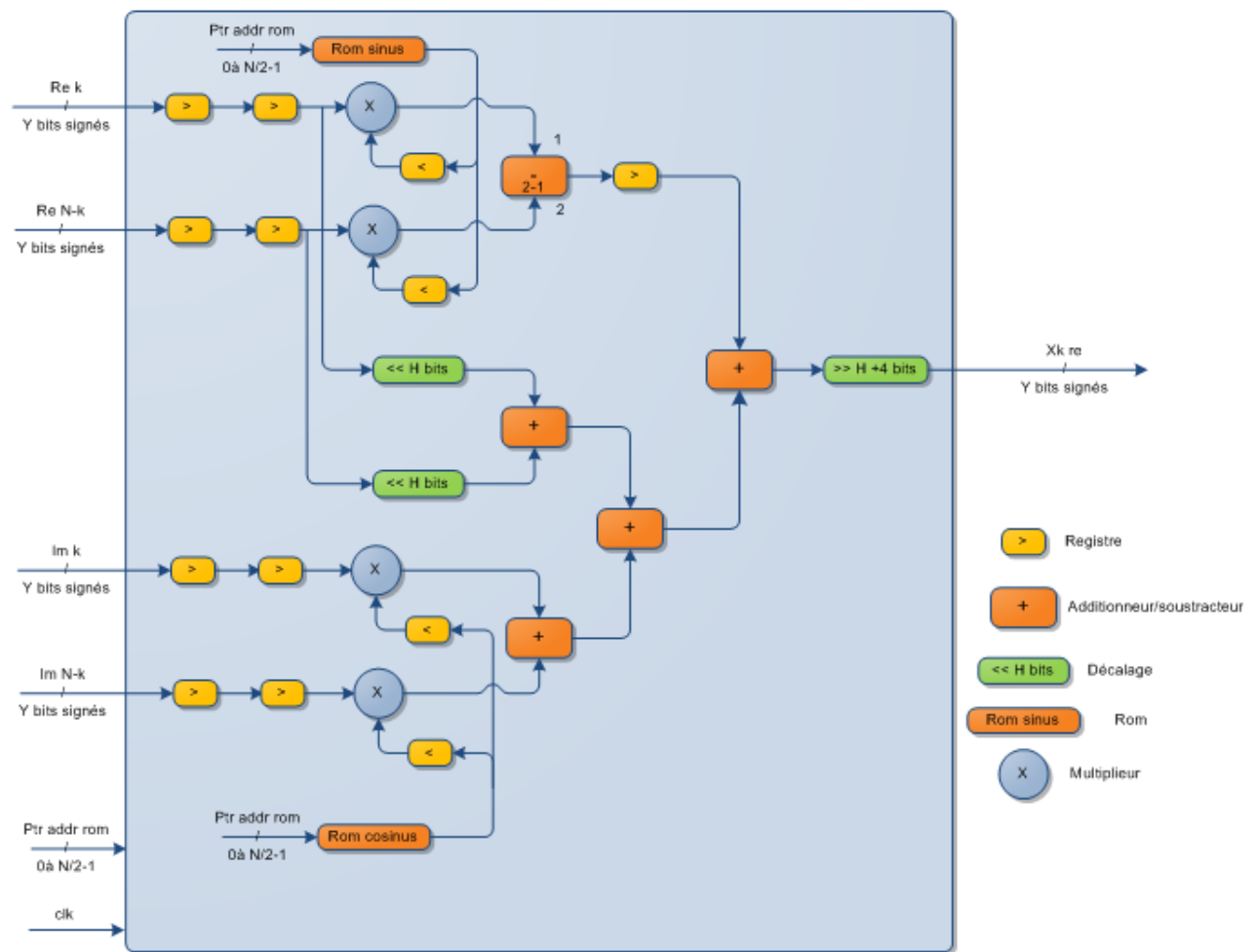


FIG. 14 – Reconstitution réelle

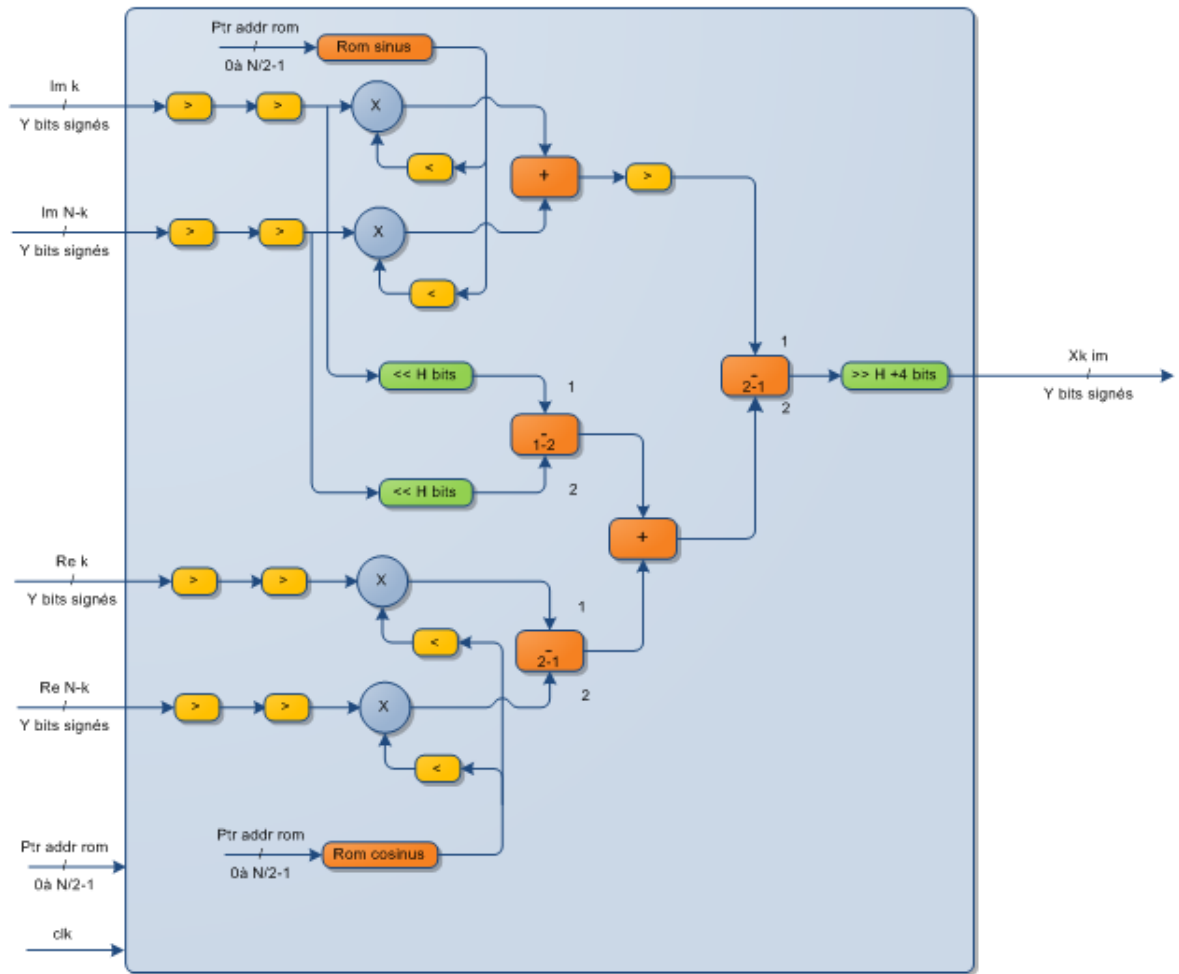


FIG. 15 – Reconstitution imaginaire

Du fait de la complexité du bloc et de la similarité des traitements pour récupérer la partie réelle et la partie imaginaire du spectre, nous le décomposerons en 2 sous-blocs. Un sous bloc de reconstitution de la partie réelle, et un autre de reconstitution de la partie complexe.

D'un point de vue architectural, il n'est pas possible de tenir la fréquence de $F_e/4$ pour le calcul de reconstitution en un cycle d'horloge. Il est dès lors nécessaire de pipeliner le calcul. Dans un premier temps, les entrées sont registrées sur 2 étages afin de garantir la cohérence des données d'entrée avec les données en sortie de rom.

Pour pipeliner le calcul, on réalise sur un premier étage les 4 multiplications en parallèle suivi d'un arbre d'addition/soustraction. Le registre en sortie de multiplieur à été volontairement « oublié » car il permet à l'outil de synthèse d'utiliser l'accumulateur du DSP48E⁸ pour réaliser l'addition ou soustraction qui suit.

Une remise à l'échelle termine le calcul.

Pour connaître l'indice de l'échantillon de sortie dans le spectre, on génère un compteur de sortie.

La latence du calcul est de 8 cycles d'horloge.

3.3 Architecture de la voie courte d'analyse N1 points

Les éléments constitutifs de cette voie ont été présentés dans la partie précédente. Il ne reste maintenant plus qu'à les assembler et les synchroniser les uns aux autres. Ce travail est grandement simplifié car chaque bloc élémentaire contient un bloc de gestion, il fonctionne donc de manière autonome après son initialisation.

Pour la synchronisation des blocs, il n'y a alors qu'à gérer une synchronisation des initialisations.

Pour simplifier la schématisation de la voie d'analyse N1, on va regrouper les blocs tampon FFT N1, FFT N1, tampon reconstitution N1, reconstitution N1 dans un bloc appelé bloc FFT N1 :

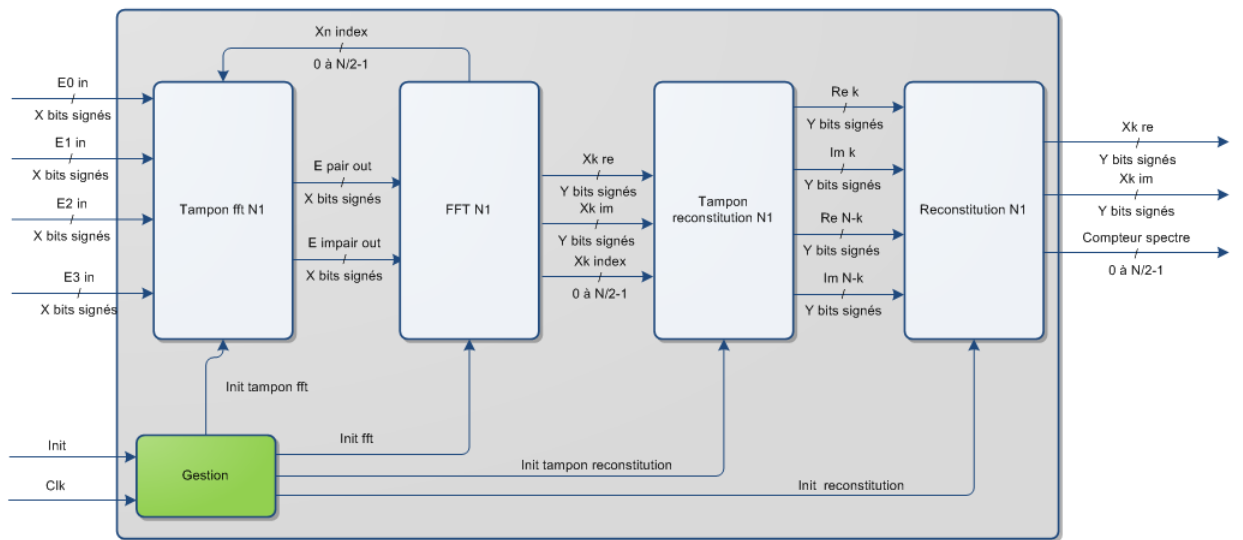


FIG. 16 – Bloc FFT N1

⁸DSP48E : multiplieurs/accumulateurs sur le virtex 5

Ce bloc est très simple, après son initialisation il réalise le traitement des $N1$ premières données sur $2.N1$ données qui lui sont envoyées. Le bloc de gestion est là pour initialiser correctement les différents blocs, en tenant compte de la latence respective de ceux-ci.

Pour traiter la totalité des données, il faut donc mettre 2 de ces blocs en parallèle et décaler leur traitement de $N1$ échantillons, soit $N1/4$ cycles d'horloge.

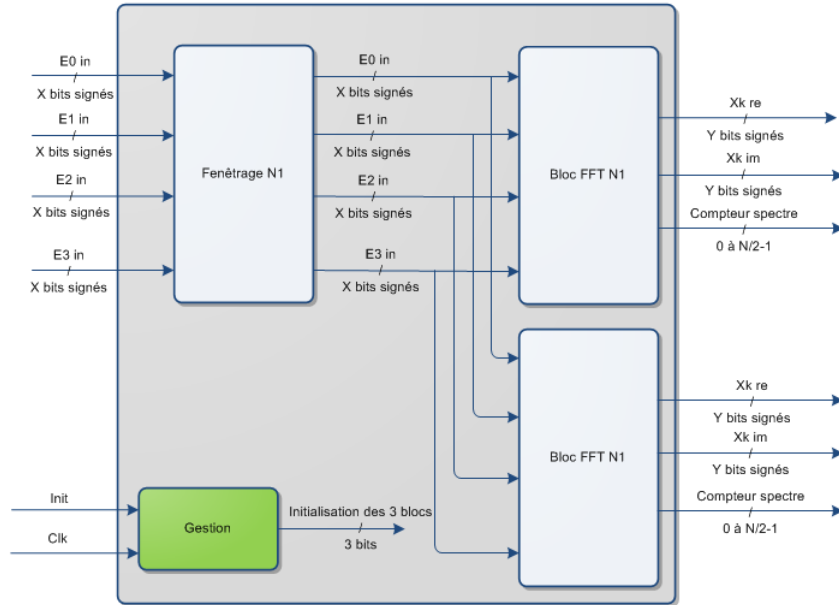


FIG. 17 – Overlap 0 de la voie d'analyse $N1$ points

La figure ne représente que l'overlap 0 de la voie d'analyse $N1$ points, pour obtenir la voie d'analyse $N1$ complète, il faut rajouter l'overlap 1, donc la fenêtre de pondération est décalée de 50% de la période. La partie gestion gère l'initialisation des différents blocs. Ainsi, il y a un décalage de $N1/2$ points soit $N1/8$ cycles d'horloge entre l'initialisation des 2 blocs de fenêtrage. Les blocs FFTs sont ensuite initialisés en conséquence et en prenant compte de la latence des blocs de fenêtrage. Chaque bloc FFT traite $N1$ échantillons sur $2.N1$ échantillons présentés.

3.4 Architecture de la voie longue d'analyse $N2$ points

Cette voie est quasi-identique à la voie $N1$ points. Les seules différences se situent au niveau de la taille des mémoires et de la taille des bus en sortie de FFT car la dynamique est différente.

3.5 Evaluation du module sous Matlab et simulation modelsim

Afin de vérifier le bon fonctionnement du design, nous allons mettre en place une procédure de test. Avec matlab, nous allons générer un signal de test qui sera exporté au format texte. Le test bench va alors prendre comme stimuli sur les entrées données le signal stocké dans le fichier texte, et exporter les résultats dans des fichiers textes de sorties. En parallèle de ce traitement nous réalisons la transposition T/F de ce même signal avec matlab. Il est ainsi possible de comparer le résultat fourni par matlab avec le résultat de l'implémentation et donc de détecter rapidement les erreurs et les corriger.

Pour améliorer la lisibilité lors des tests, nous ne représenterons que les demi-spectres, le signal d'entrée étant en effet réel, le spectre de sortie sera symétrique.

Le signal de test sera composé de sinusoïdes à différentes fréquences, qui dans le cas idéal d'une transformée de Fourier ferait apparaître des diracs dans la représentation fréquentielle, aux fréquences correspondantes des sinusoïdes. Dans notre cas, la transformation n'étant pas idéale, nous devrions voir apparaître des pics à ces fréquences.

3.5.1 Test de la voie courte N1 points

Supposons qu'une trame de $N1$ points dure une période T , en générant un signal qui se répète 2 fois dans la trame, on doit voir apparaître un pic dans le domaine fréquentiel à l'indice 2 du spectre.

En générant un signal composé de sinusoïdes de fréquence $2/T, 15/T$ et $20/T$, des pics doivent apparaître en sortie d'analyse spectrale aux indices 2, 15 et 20.

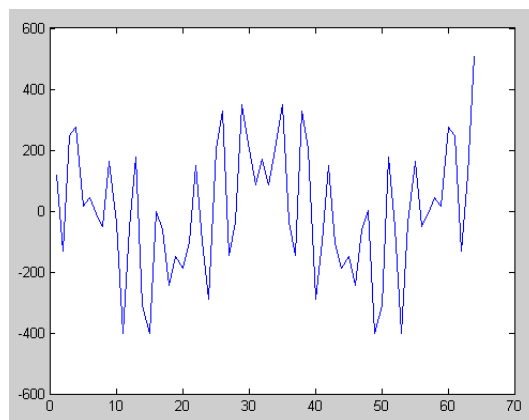


FIG. 18 – Signal à transposer dans le domaine fréquentiel

Calcul de sa FFT N1 points avec matlab (le signal à été préalablement pondéré par une fenêtre de pondération pour obtenir le même résultat qu'avec l'implémentation) :

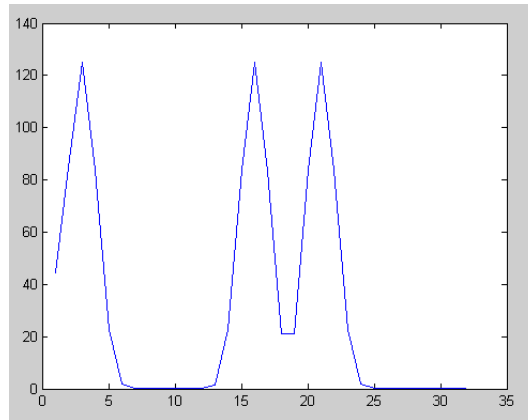


FIG. 19 – Module de la FFT N1 points calculée sous Matlab

A comparer au résultat obtenu par la simulation du module sous Modelsim :

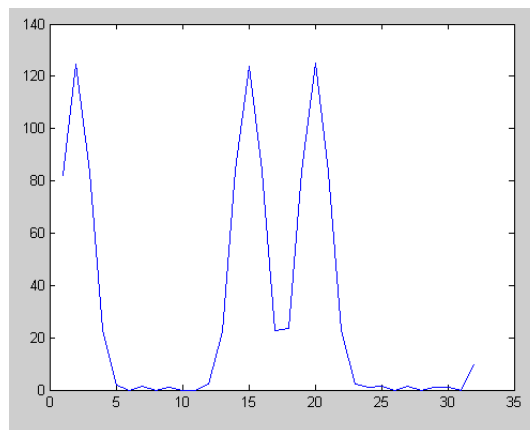


FIG. 20 – Module de la FFT N1 points simulée sous modelsim

Pour comparer les résultats, l'écart de dynamique a été compensé (ie les résultats matlab ont été divisés par 8) , on obtient ainsi l'erreur suivante en prenant pour référence le calcul matlab :

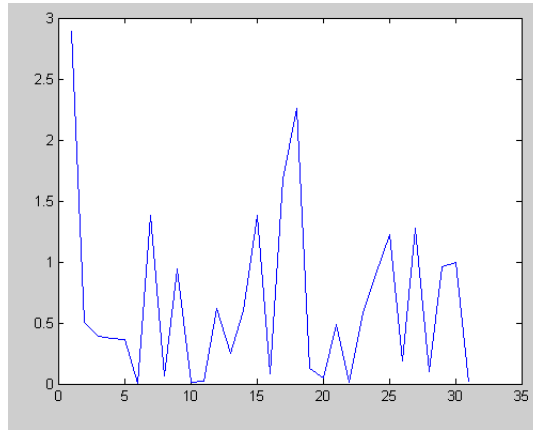


FIG. 21 – Erreur absolue entre le calcul et le modèle simulé

L'erreur absolue est inférieure à 2.5% de la valeur maximale de la densité spectrale. Ceci n'est qu'un calcul particulier, mais en règle générale, l'erreur ne dépasse pas les 3%. Cette erreur est essentiellement due à la troncature des résultats lors de l'implémentation (limitation de dynamique des cosinus, sinus, troncature des résultats de multiplication,...)

Il est possible d'obtenir un résultat plus précis, au prix d'une augmentation des ressources occupées, déjà limites pour notre implémentation. Nous considérerons alors cette erreur comme acceptable.

3.5.2 Test de la voie courte N2 points

Pour réaliser la comparaison des résultats, nous allons utiliser le même signal que pour le test de la voie 64 points, mais répété 8 fois pour former une trame de 512 points. On comprend alors que dans ce cas, les pics de densité spectrale doivent apparaître aux index 16,120 et 160.

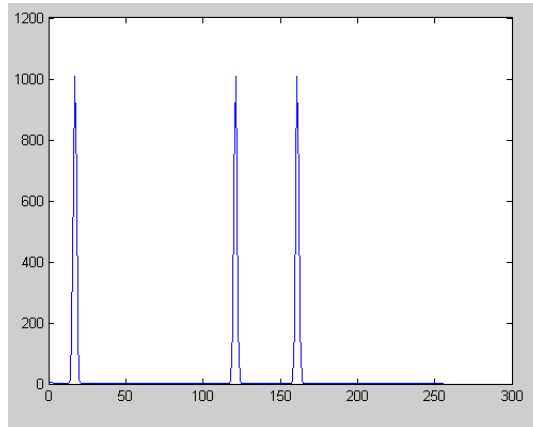


FIG. 22 – Module de la FFT N2 points calculée sous Matlab

L'erreur est ici de 0.5 %. En effet, le fait d'effectuer une FFT plus longue augmente la dynamique de sortie de celle ci, et rend donc le calcul moins sensible au bruit de troncature. On constate un rapport de quasiment $N2/N1$ entre l'erreur de la voie N1 et l'erreur de la voie N2.

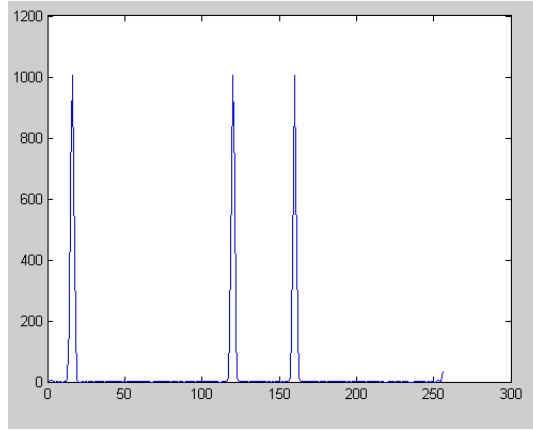


FIG. 23 – Module de la FFT N2 points simulée sous modelsim

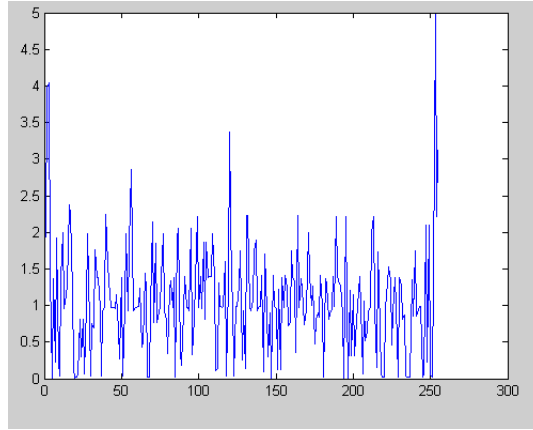


FIG. 24 – Erreur absolue entre le calcul et le modèle simulé

Les tests ci-dessus ont montré la fonctionnalité du module. Nous avons montré que l'erreur absolue de calcul restait raisonnable sur les 2 voies d'analyse. (de 3% pour la voie N1 points, à 0.5% pour la voie N2 points). Une simple augmentation de la dynamique de sortie du signal pourrait réduire cette erreur (car le signal est tronqué en sortie), car elle est en grande partie due à la troncature des données dans le calcul, mais aussi sur la sortie.

Nous sommes de plus dans le cas idéal où toute la dynamique d'entrée est utilisée, l'erreur de calcul augmentera lorsque toute cette dynamique ne sera pas utilisée. Pour une meilleure précision dans le calcul, il faudra donc une adaptation de gain en amont de l'analyse spectrale.

3.5.3 Etude de la répétitivité et de la cohérence des différentes voies

Nous avons montré la fonctionnalité du module uniquement sur les premières données fournies, et sur une seule chaîne fenêtrage et bloc FFT.

Il faut maintenant s'assurer que les différentes voies fournissent les mêmes résultats (cohérence) et que pour des données identiques en entrées, les résultats restent les mêmes (répétitivité).

Pour ce test, on utilise le même signal d'entrée que pour les tests précédents.

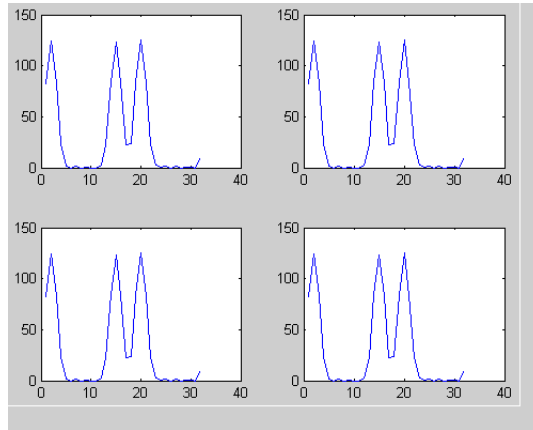


FIG. 25 – Cohérence de la voie N1 points

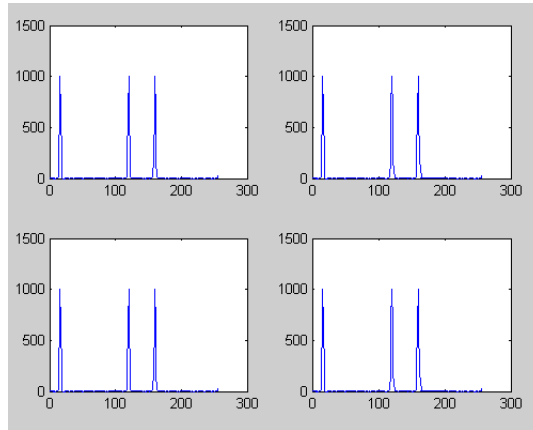


FIG. 26 – Cohérence de la voie N2 points

Les 2 voies de l'overlap 0% et les 2 voies de l'overlap 50% sont représentées sur les figures. Les FFTs N1 et N2 points sont bien identiques sur les 4 voies, elles sont donc bien synchronisées entre elles.

3.5.4 Etude de la répétitivité

Pour vérifier le bon fonctionnement du système, il faut aussi s'assurer que pour un même signal présenté en entrée, les spectres restent identiques et qu'il n'y ait pas d'arrêt du système, ou de décalages d'indices au cours du temps.

Nous allons donc comparer 2 spectres d'une même voie, pris arbitrairement parmi les différents spectres calculés.

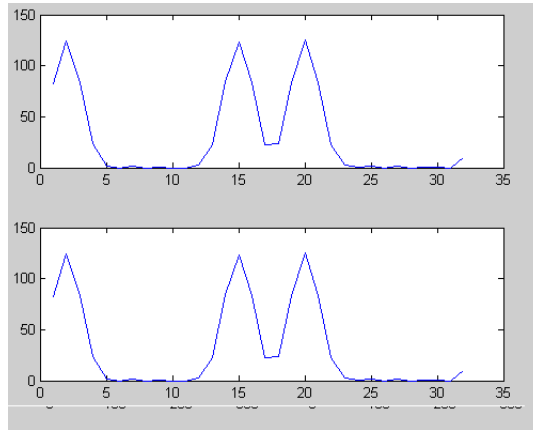


FIG. 27 – Test de répétitivité voie N1 points

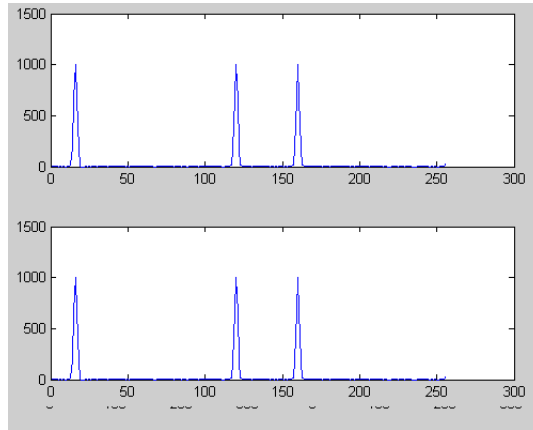


FIG. 28 – Test de répétitivité voie N2 points

On peut donc considérer que les traitements réalisés sur les signaux d'entrées sont les mêmes pour les 2 spectres, et donc qu'il y a bien répétitivité du calcul.

3.6 Conclusion des travaux sur l'analyse spectrale

Les résultats montrent la faisabilité d'une telle chaîne de transposition T/F sur les FPGAs actuels et futurs. Les contraintes architecturales se révèlent tout de même assez fortes, du fait des fréquences de fonctionnement élevées, atteignant presque les limites des modules élémentaires. La densité du design augmente de plus la difficulté de le placer/router sur le FPGA.

Une telle solution rend véritablement obsolète l'utilisation de DSP dans une application similaire, on réalise un véritable bon en avant au niveau des performances, et nul doute que ces

solutions seront utilisées de plus en plus largement pour des applications de traitement du signal.

Ces travaux ont pu être validés au travers du stage de Sébastien VALLET et de Stéphane JOIE, également étudiants de l'ENSIETA en stage de 2 ième année dans le service. Leur stage s'intéressait à la reconfiguration dynamique des FPGAs, nous en avons profité pour réaliser une application utilisant la reconfiguration dynamique de la chaîne d'analyse pour avoir au choix une FFT voie courte ou une FFT voie longue. Le design complet n'est alors pas totalement testé, mais la partie la plus critique l'est et s'est révélée fonctionnelle.

4 Pile UDP/IP Gigabit Ethernet sur FPGA

Nous l'avons vu dans la partie précédente, le débit d'information sortant de l'analyse spectrale est très soutenu. Des blocs de traitement de l'information et de détection sont implémentés dans le système FrontEnd, traitant directement les informations en sortie d'analyse spectrale. On évite ainsi le rapatriement de toutes les données vers le système BackEnd. Il est cependant intéressant d'étudier des solutions permettant d'augmenter le débit de transmission entre les deux systèmes, car les algorithmes de traitement implémentés dans le FrontEnd doivent rester relativement simples pour pouvoir être implémentés sur le composant. Il peut donc être intéressant de disposer d'une liaison haut débit entre les deux systèmes, pour transmettre le maximum d'informations, et pouvoir les traiter à posteriori avec des algorithmes plus complexes. Cela permet de mieux identifier la cible, de mieux s'y préparer et également d'orienter le développement des algorithmes de traitement embarqués.

Dans le cas d'utilisation de coprocesseur pour le traitement numérique, il est également intéressant de disposer d'une liaison haut débit entre la carte de traitement et le système car ce sont en général les débits de communication qui brident les systèmes de calculs distribués.

Jusqu'alors cantonné au bus VME, les transmissions entre les systèmes FrontEnd et BackEnd ne se satisfont plus de cette technologie, une augmentation du débit de transmission est nécessaire. L'étude d'une liaison Gigabit Ethernet entre les 2 systèmes a donc été menée, ayant pour conséquence une augmentation considérable du débit et de nouvelles possibilités d'interconnexions grâce à la norme Ethernet, communément utilisée dans le monde informatique.

4.1 Mise en situation

4.1.1 Présentation

La norme Gigabit Ethernet[2] définit un standard de communication haut débit, compatible avec les anciennes générations de liaisons Ethernet. Dans ce type de liaison, l'encapsulation et la désencapsulation des données sont généralement réalisées en software par le ou les processeurs de la carte. Dans le cas d'une liaison Gigabit Ethernet, le débit des données devient tel, qu'il est préférable d'adopter une méthode d'encapsulation et de désencapsulation exclusivement hardware. Cela permet ainsi de libérer du temps CPU pour les applications plus critiques. Un tel module permet aussi à n'importe quelle carte électronique de devenir télé-opérée, à travers un simple réseau informatique.

Le projet est donc de réaliser une pile UDP/IP capable d'encapsuler, de désencapsuler les données UDP, et de s'identifier sur un réseau informatique à l'aide des protocoles communément utilisés [1].

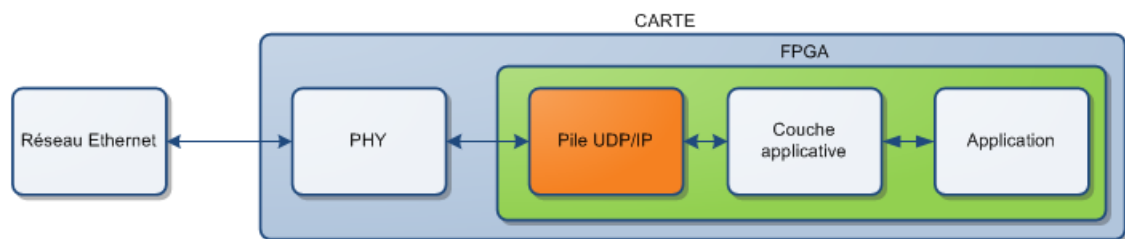


FIG. 29 – Situation de la pile UDP/IP dans le système

La pile UDP/IP fait le lien entre la couche physique (très bas niveau) et la couche applicative (haut niveau) du système, et s'avère donc critique pour les performances et la fiabilité du système.

Le système doit pouvoir s'identifier sur un réseau Ethernet, ce qui implique de gérer les protocoles d'identification ARP⁹ et ICMP¹⁰.

Une couche applicative de niveau supérieur à UDP est nécessaire pour rendre transparente l'utilisation du protocole réseau pour l'application, cette couche sera simplement une encapsulation à définir au cas par cas.

4.1.2 Performances attendues

Comme dans toute communication encapsulante, le débit utile est toujours inférieur au débit de la liaison. Plaçons-nous dans le cas le plus favorable, lorsque le transfert est continu, on utilise alors la taille maximale admissible pour les trames IP : 1500 octets. Il faut rajouter à cette trame le préambule de 12 octets, le checksum de 4 octets et le délai inter-trame équivalent à 8 octets.

L'entête Ethernet se compose de 14 octets, l'entête IP de 20 octets et l'entête UDP de 8 octets, le débit utile en streaming passe alors à :

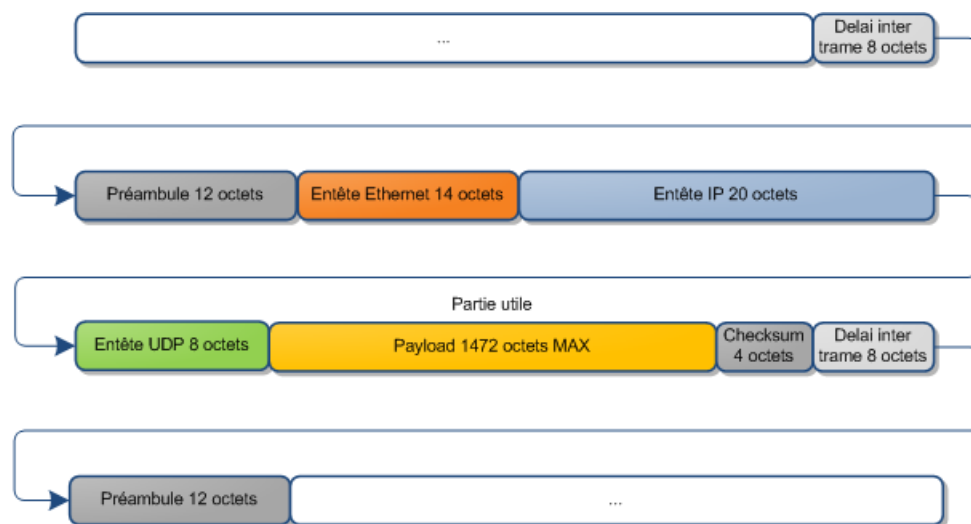


FIG. 30 – Encapsulation des données dans le cas d'un streaming

$$(1500 - 20 - 8) / (1500 + 14 + 12 + 4 + 8 + 20 + 8) \cdot 10^9 = 939.97 \text{ Mbps soit } 117.5 \text{ Mo/s.}$$

La pile UDP/IP traite les informations en flot continu (streaming), ce débit est donc atteignable dans le cas d'une architecture optimisée.

⁹ARP : Protocole permettant de faire la correspondance entre une adresse MAC et une adresse IP

¹⁰ICMP : Protocole de diagnostic réseau, il permet de détecter des problèmes sur le réseau. Dans notre cas, seule la partie ping du protocole sera supportée, elle permet de connaître l'état d'une machine sur le réseau et son temps de réponse

4.2 Architecture du module

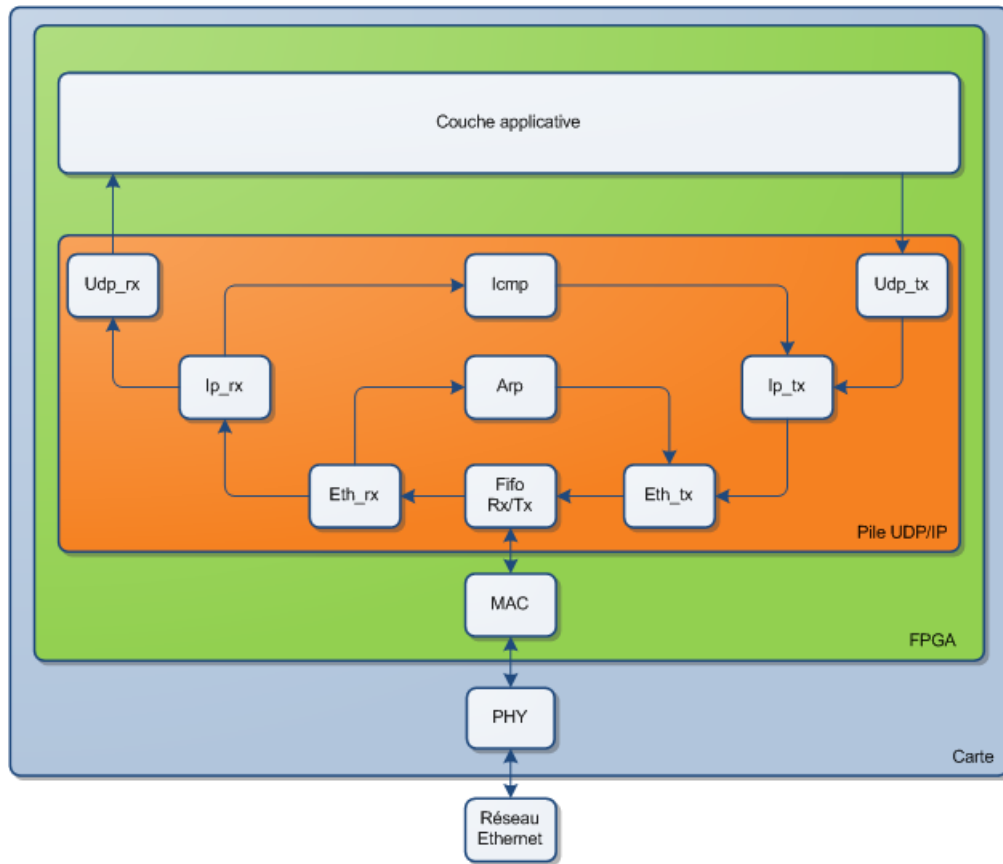


FIG. 31 – Architecture de la pile UDP/IP

Le synoptique ci-dessus représente l'architecture du module à travers les différents sous-modules. Afin de simplifier la validation et l'évolution du module, celui-ci est dissocié en différents modules élémentaires. Cette approche permet de valider un à un les différents sous-modules, et par la même de valider le module de façon progressive.

La pile UDP/IP a été conçue dans un but de réutilisation aisée, il suffira de faire appel à cette brique élémentaire pour disposer d'une connectivité Gigabit Ethernet sur le FPGA, indépendamment du modèle visé. Son interfacage avec les éléments extérieurs est simple. La configuration de la pile se fait par un port de configuration qui comprend l'adresse MAC, l'adresse IP, le port UDP source, le port UDP client, ainsi que d'autres paramètres de configuration plus précis.

Afin de faciliter le débogage, un système de détection d'erreur a été mis en place, permettant de localiser l'origine et l'instant de l'erreur. Il est alors possible de connaître quel sous-module

génère une erreur ainsi que la cause de cette erreur.

Chaque sous module étant architecturé autour d'une machine d'état, la principale difficulté de la pile réside dans la synchronisation de toutes ces machines entre elles.

Voici une description succincte des différents sous-modules de la pile :

MAC Xilinx : Ce module gère la couche bas niveau de la communication Ethernet. Il réalise en réception :

- la génération d'horloge à partir du préambule de la trame
- le filtrage d'adresse MAC
- la vérification du checksum

Et en émission :

- l'insertion de préambule et de checksum dans la trame

Ce module est une IP Xilinx qu'il suffit de configurer suivant les besoins de l'application.

Fifo rx/tx : Ce module sert de mémoire tampon pour la réception et l'émission de trames, seules les trames valides en réception y sont stockées. Il réalise de plus le découplage des domaines d'horloges, car les parties réception et émission du MAC fonctionnent avec des horloges asynchrones. Cela permet de s'affranchir le plus tôt possible des 2 domaines d'horloge, en faisant fonctionner le reste du design dans un même domaine d'horloge. Une interface Locallink¹¹ est mise en place pour la partie client du module.

Eth rx : Ce module sert à filtrer les adresses MAC et aiguille les données vers les protocoles correspondants (protocoles reconnus : IP et ARP). Les données sont ensuite transmises au module du protocole associé.

Ip rx : Ce module vérifie l'intégrité de l'entête IP, vérifie l'adresse IP destinataire et calcule le checksum de l'entête. Une fois ces conditions réunies il peut aiguiller les données vers le module associé au protocole de niveau supérieur. Pour l'instant ne sont reconnus que les protocoles UDP et ICMP.

Udp rx : Ce module réalise un découplage du domaine d'horloge entre la partie pile UDP/IP et la partie applicative. La partie applicative doit en effet pouvoir fonctionner à une fréquence différente des 125 Mhz nécessaires au Gigabit Ethernet. Il réalise ainsi un tamponnage des données à l'aide d'une fifo asynchrone, et adapte aussi les interfaces de communications.

Arp : Ce module sert à l'identification du système sur le réseau. Il sert à faire la correspondance entre l'adresse IP d'une machine et son adresse MAC. En réception, il identifie s'il s'agit d'une requête ou d'une réponse, dans le cas d'une requête, il demande au bloc eth tx l'autorisation d'émettre une réponse ARP. Il stocke au passage l'adresse MAC correspondante à l'adresse IP de la machine dans sa table ARP d'une capacité de 8 couples MAC/IP. Dans le cas d'une réponse,

¹¹Locallink : Interface de communication simple utilisée dans les IPs Xilinx : un bus de data et 4 signaux de contrôle

il met également à jour sa table ARP. Le module eth tx interroge le module ARP lorsqu'il veut effectuer une transmission, pour connaître l'adresse MAC du destinataire. Si celle-ci est stockée dans la table ARP, le module ARP renvoie directement l'adresse MAC, sinon il demande au bloc eth tx l'autorisation d'émettre une requête ARP. Lorsque le module ARP disposera de l'adresse MAC du destinataire, il informera le module eth tx. Un timeout est également nécessaire, en cas de non-réponse du destinataire à une requête ARP, pour éviter de bloquer le système dans un état d'attente.

Eth tx : Ce module sert à encapsuler les données des couches supérieures. Pour connaître les adresses MAC de destination, une correspondance est demandée auprès du bloc ARP. Le module ARP dispose d'un niveau de priorité plus important à l'émission.

Ip tx : Ce module sert à encapsuler les données des couches supérieures, sont supportés à présent les protocoles ICMP et UDP. Il calcule le checksum de l'entête IP et l'insère dans la trame. Pour transmettre les données au module eth tx, il émet une requête à celui-ci, en indiquant l'adresse IP du destinataire, et n'est autorisé à transmettre ses données que lorsque le destinataire est identifié sur le réseau par son adresse MAC. La procédure de transmission peut être abandonnée en cas de dépassement du timeout du module eth tx. Dans ce cas, la notification d'erreur est remontée au niveaux supérieurs pour que la transmission de la trame soit abandonnée.

Icmp : Ce module sert à tester le réseau. Il permet de répondre à une requête ping. Il est donc en lien direct avec les modules ip rx et ip tx.

Udp tx : Ce module permet d'encapsuler les données UDP et les transmet au module ip tx. Il joue aussi le rôle de mémoire tampon entre la couche applicative et la partie transmission qui peuvent fonctionner à des fréquences différentes.

Il faut noter que les modules Eth rx, Ip rx, Ip tx et Eth tx sont purement des blocs d'aiguillage, ils ne réalisent aucune fonction de mémorisation des données. Celles-ci sont transmises dès qu'elles sont validées par le bloc amont.

4.3 Validation

La validation d'un module de cette complexité nécessite un effort conséquent. Les premiers tests permettent de vérifier un fonctionnement de base et valident l'architecture retenue. Ils sont encourageant pour la suite du travail, mais la validation totale du module ne peut se faire qu'à partir d'une batterie de tests définis dans la spécification du module. L'objectif est de s'assurer que dans l'ensemble de tests choisis, représentatifs du fonctionnement en situation réelle, les machines d'états ne se figent pas et que tout fonctionnement anormal est signalé par une erreur.

La validation s'est faite par étapes, en exploitant l'architecture modulaire de la pile. Chaque couche a été testée et validée avant la conception de la couche suivante.

Le module fonctionne actuellement sur une carte d'évaluation à base de FPGA Spartan 3A

DSP dans l'attente de la carte définitive en cours de production. Il est encore en phase de test de robustesse, mais le fonctionnement global est validé. Nous le verrons dans la suite du rapport, ce module est utilisé pour l'amélioration d'une carte de traitement numérique.

4.4 Ajout de la connectivité Gigabit Ethernet à une carte de traitement numérique

L'étude de la pile UDP/IP permet à Thales d'avoir un module de communication Gigabit Ethernet prêt à l'emploi, mais a également été mené dans le but d'améliorer la flexibilité et les performances en communication d'une carte de traitement déjà présente dans plusieurs systèmes.

La carte de traitement permet de réaliser des calculs répétitifs et à très fortes contraintes temporelles. Pour cela, de gros FPGAs sont embarqués sur la carte, et sont chargés par des firmwares dépendants de l'application visée. Actuellement les FPGAs opérationnels communiquent uniquement avec le reste du système par un bus VME via un FPGA de servitude. Les débits et les contraintes matérielles de ce type de bus ont amené les architectes numériques à choisir un autre mode de communication : l'utilisation associée du bus VME pour le contrôle des cartes et de réseaux Ethernet pour les flots de données permet une plus grande flexibilité et évolutivité du système, améliorant au passage les performances globales du système.

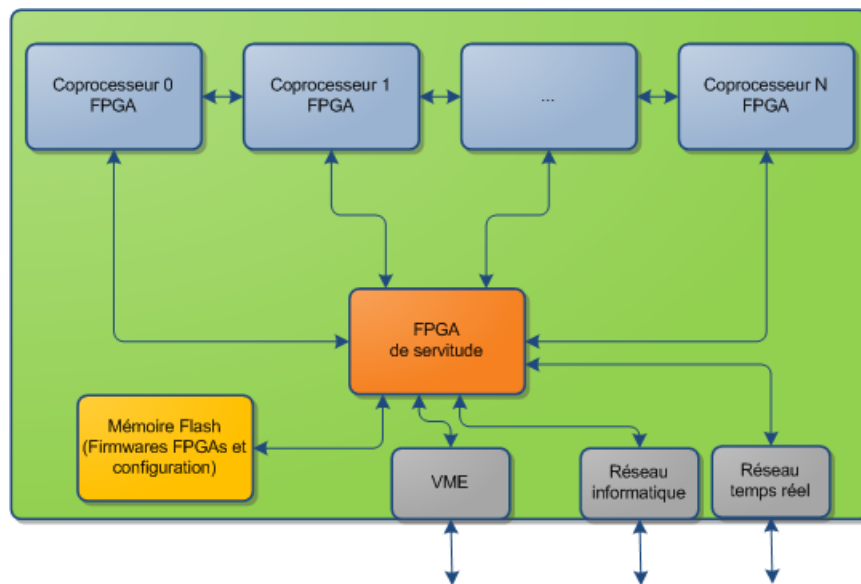


FIG. 32 – Architecture de la carte de traitement numérique

J'ai participé au processus de développement de cette amélioration, en ayant à ma charge le firmware du FPGA de servitude. La principale modification est l'ajout de deux ports de connexion Gigabit Ethernet. Un de ces port est connecté au réseau informatique global, sans réelles contraintes temporelles. L'autre est utilisé pour le temps réel, le débit des informations et le temps de réponse est alors critique.

Les FPGAs doivent pouvoir communiquer avec les 2 ports Ethernet pour recevoir et envoyer des données. Or ils ne disposent que d'un bus local de communication avec le FPGA de servitude. Un système de routage des paquets UDP et de partage du port physique a été mis en place.

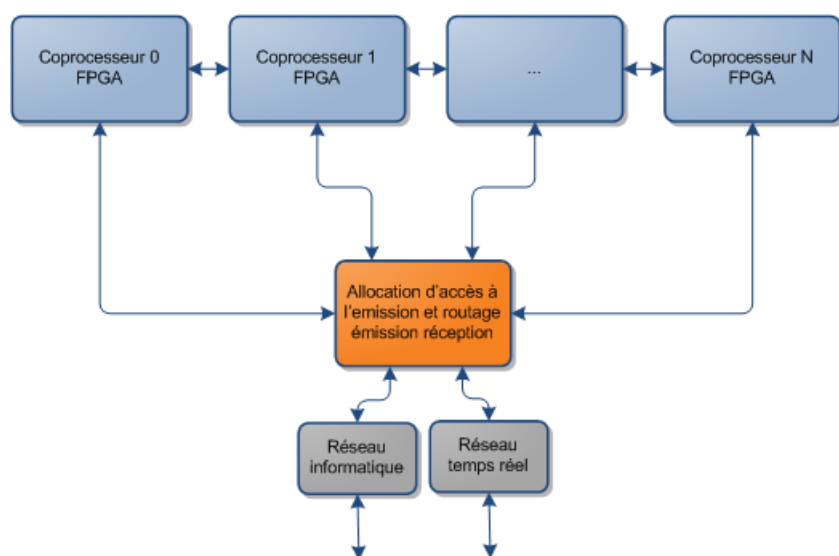


FIG. 33 – Routage des packets et partage de l'accès

Pour l'allocation d'accès en émission une règle simple a été utilisée : lorsque plusieurs FPGAs veulent émettre sur le même port physique, le FPGA ayant eu la main il y a le plus longtemps la reprend.

Ces évolutions sont en cours de réalisation(simulation fonctionnelle). Dès l'arrivée de la carte définitive (mi-septembre), je participerai à sa validation, en m'appuyant sur la procédure en cours d'écriture à laquelle je contribue.

5 Conclusions et perspectives du PFE

Ces deux exemples illustrent parfaitement la montée en puissance des FPGAs dans les systèmes orientés traitement du signal ainsi que communication. Les matrices atteignent des tailles telles que désormais on peut considérer un FPGA comme un véritable système, entièrement reconfigurable. Il n'est pas rare en effet de retrouver des micro-processeurs embarqués, des modules de communication, des coprocesseurs mathématiques... De nombreuses possibilités s'ouvrent aux ingénieurs, et de nouveaux outils de synthèse de haut niveau voient depuis quelques années le jour. Les possibilités de créations deviennent telles que les outils dont nous nous servions il y a quelques années ne nous permettent plus de toutes les explorer. Ainsi des logiciels se basant sur du code C adapté aux processus parallèles permettent dès aujourd'hui de créer des systèmes sur puce. Une version de l'analyse spectrale basée sur ces outils est d'ailleurs en cours d'évaluation dans le laboratoire, et permettra à terme, de modifier les choix d'architecture du design en quelques minutes...

Ce stage était donc pour moi l'opportunité de découvrir le métier d'ingénieur en électronique numérique et de me conforter dans mes choix d'orientation. Cette expérience acquise à travers les 12 mois passés au sein de l'entreprise m'a permis de découvrir le cycle de développement d'une carte électronique d'un grand groupe industriel, de me former sur mon métier au travers le dialogue et l'échange d'idées avec les ingénieurs du service. Le contrat de professionnalisation est pour moi un véritable plus dans la formation d'un ingénieur, car il permet de découvrir l'entreprise, de se former dans la première partie de l'année, pour être opérationnel et plus autonome lors de la période de stage de fin d'étude. C'est dans cette seconde période que j'ai pu véritablement mettre à profit l'ensemble de ces acquis et découvrir la richesse du métier. C'est pourquoi je le conseillerai aux étudiants désireux de rentrer de plein-pied dans le monde de l'industrie.

Mais cette année dense se révèle surtout très riche sur le plan personnel d'un point de vue formation technique et humaine. Les connaissances acquises lors de la formation à l'ENSIETA, qui garantissent une bonne approche de la situation, se transforment en compétences professionnelles au fil de notre travail et de notre implication dans le service. Mon travail a énormément gagné en autonomie et en rigueur, les conséquences sont en effet toutes différentes entre un projet scolaire et un projet industriel. Le côté humain est tout aussi important dans la réussite de ce stage. L'équipe sympathique et chaleureuse m'a permis de passer de très agréables moments dans le service et d'atteindre les objectifs que je m'étais fixé pour le stage. Je sais à présent vers quel domaine je vais orienter ma carrière professionnelle...

A Présentation de Thales et du service architecture numérique

Présent dans 50 pays et employant 68 000 collaborateurs, Thales est leader mondial des Systèmes d'information critiques sur les marchés de l'Aéronautique et de l'Espace, de la Défense et de la Sécurité.

En maîtrisant les grands systèmes logiciels, Thales répond aux défis de sécurité que ses clients ont à relever dans un monde toujours plus complexe qui fait appel à des technologies de plus en plus sophistiquées.

Les systèmes civils et militaires partagent de nombreuses innovations. Thales est reconnu pour sa longue expérience du développement en synergie des technologies duales impliquant un réseau mondial de 22 000 chercheurs de très haut niveau.

Fort de sa dimension internationale, et de sa capacité à couvrir l'ensemble de la chaîne de valeur, de la maîtrise d'oeuvre aux équipements, Thales s'affirme comme l'acteur clé d'un monde plus sûr.

cf : <http://www.thalesgroup.com>

J'effectue mon stage au sein du Service Architecture Numérique(SAN) de la division aéronautique de Brest. Le service est intégré dans le département Ingénierie et Intégration Matérielle(I2M) dont la mission consiste en la conception, le développement et la mise au point de plateformes, de sous-ensembles matériels et de logiciels de base entrant dans la constitution des systèmes de guerre électronique Thales.

34 personnes participent à la vie du service (24 ingénieurs et cadres, 4 techniciens, 1 secrétaire, 4 apprentis et 1 Thésard), réparties suivant trois pôles :

- Le pôle programme
 - responsabilité d'affaires
 - gestion de lots
- Le pôle ingénierie
 - Architectures calculateurs à base de COTS(Commercial Off The Shelf)
 - Conception de cartes et de sous-ensembles périphériques
 - Intégration en coffret des conceptions
 - Développement, intégration du logiciel de base
- Le pôle CLP(Composants Logiques Programmables)
 - Développement FPGA
 - Déclinaison d'algorithmes provenant de la DT

Le domaine d'activité de SAN est essentiellement tourné vers les activités propres au centre de Brest : de la guerre électronique et de la patrouille maritime(calculateurs, traitement du signal radar...)

Références

- [1] FrameIP. Description des protocoles utilisés en ethernet. www.frameip.com/.
- [2] IEEE. Ieee 802.3. <http://www.ieee802.org/3/>.
- [3] Numerical Recipes. Méthodes de calcul numérique. <http://www.nr.com/>.
- [4] Wikipedia. Définition d'un fpga. http://en.wikipedia.org/wiki/Field-programmable_gate_array.
- [5] Wikipedia. Histoire de la guerre electronique. <http://fr.wikipedia.org/wiki/GuerreElectronique>.