

Implantation d'un démodulateur numérique sur FPGA

Emmanuel Boutillon, Jean-Luc Danger

Ecole Nationale Supérieure des Télécommunications,
46 rue Barrault,
75634 PARIS CEDEX 13, FRANCE

RÉSUMÉ

La nouvelle génération de circuits logiques programmables permet l'intégration de systèmes complexes et hétérogènes. Mais, la structure figée de leur architecture modifie l'expertise de l'architecte de circuit. Le problème l'interaction entre algorithme et architecture se traduit différemment dans les domaines des CLP et des ASIC.

Dans cet article, nous montrons comment la prise en compte à différents niveaux des caractéristiques du FLEX10K50 de ALTERA a permis la réalisation d'une fonction de démodulation complexe. La structure des cellules élémentaires nous a conduit à définir une architecture de multiplieur efficace (Booth modifié). Le plan de masse du FLEX10K50 a orienté le choix de l'algorithme de démodulation. Enfin, le manque de ressources matérielles nous a obligés à définir un nouvel algorithme de synchronisation..

1 Introduction

Les nouvelles générations de circuits logiques programmables (CLP) offre l'équivalent de plusieurs dizaines de milliers de portes ainsi qu'un certain nombre de bancs mémoires. Elles ouvrent ainsi la possibilité d'intégrer un système complexe sur un unique CLP, et ce, avec un coût de développement et un temps de disponibilité sur le marché inférieur à ceux d'une filière ASIC. Mais, à la différence de cette dernière pour laquelle le concepteur a un grand nombre de libertés sur sa conception (nombre de portes, type d'architecture), un CLP a une organisation interne figée et surtout une quantité fixe de matériel. L'intégration efficace d'une application sur un CLP doit donc tenir compte de ces contraintes à toutes les étapes de la conception, à savoir, le choix de l'algorithme, voire, leur reformulation pour obtenir une complexité compatible avec la cible technologique, la conception des opérateurs arithmétiques et enfin la dernière étape de placement routage.

Ces différents niveaux d'adéquation algorithme-architecture pour une cible CLP sont illustrés par un exemple concret : l'intégration sur un FLEX 10K50 [1] d'un démodulateur numérique pour une application de transmission sans fil. Le premier chapitre présente la méthode de démodulation. Dans le deuxième chapitre, nous montrons comment l'organisation matérielle du FLEX 10K50 a mené au choix de l'algorithme. Le chapitre suivant présente un opérateur de multiplication optimisé pour la structure du FLEX10K50. Enfin avant de conclure, nous

ABSTRACT

A new generation of Programmable Logic Devices allows the integration of complex and heterogeneous systems. But, due to the pre-defined structure of a PLD, the optimization of algorithm and architecture is different from that of classic ASIC technologies. In this paper, we give examples of optimization with the design of a demodulator using an ALTERA FLEX10K50. First, the basic cell structure of the PLD leads us to define an efficient multiplier architecture (modified Booth). Then the floorplan guides us in the choice of the filter bank architecture. Finally, the lack of sufficient on-chip resources leads to define a new algorithm for synchronization.

montrons comment le problème de recherche de synchronisation a été reformulé pour être compatible avec les ressources matérielles.

2 Principe de démodulation

Le principe de la modulation étudié est simple afin d'avoir un émetteur bas coût. La transmission d'un bit s'effectue par l'émission d'une sinusoïde pure sur la première ou deuxième moitié de la durée T de transmission d'un bit (modulation "Manchester") comme indiqué figure 1. Les données sont organisées en paquets constitués d'une séquence d'apprentissage, du message et enfin, d'un code détecteur d'erreurs (ce dernier n'est pas traité dans le circuit). Les paquets sont émis à des instants et à des fréquences (variant dans une plage de f_{max}) inconnus du récepteur.

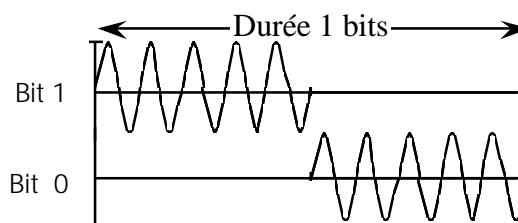


Figure 1 : Modulation Manchester

Au niveau du récepteur, une démodulation non cohérente par détection d'énergie est effectuée. le signal est échantillonné (en complexe) à une fréquence de f_{max} . La première étape de démodulation consiste à diviser la bande en 16 sous-bandes à l'aide d'un banc de filtres. Ainsi, la

sous-bande contenant le signal utile ne sera pas affectée par un éventuel brouilleur situé dans une bande adjacente. Dans chaque bande, l'énergie du signal est accumulée sur une durée d'un huitième de bit (une phase). Pour chaque bande et pour chaque nouvelle phase, le bit est décodé. Il est donné par le signe de la différence entre la somme des énergies sur les quatre premières et les quatre dernières phases (figure 2).

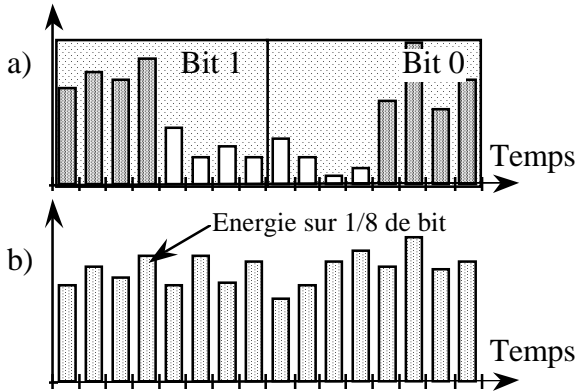


Figure 2 : Energie des échantillons
 a) dans la bande où le signal est émis (après synchronisation)
 b) dans une bande sans signal

Dans chaque bande et pour chaque phase, une fois le bit décodé, un test est effectué pour tenter de retrouver la séquence d'apprentissage dans les derniers bits reçus.

Lors de la réception d'un message, la séquence d'apprentissage se retrouve dans les bandes et les phases adjacentes. Le démodulateur doit donc évaluer la qualité des alertes puis choisir de décodé dans la phase et la bande ayant le meilleur rapport signal à bruit. Une fois celle-ci détectée, le système attendra de recevoir l'intégralité du paquet avant de rechercher une autre synchronisation. La figure 6.a présente le principe de cette démodulation.

3 Architecture du FLEX10K50 d'ALTERA

L'architecture du FLEX10K50 est représentée figure 3.

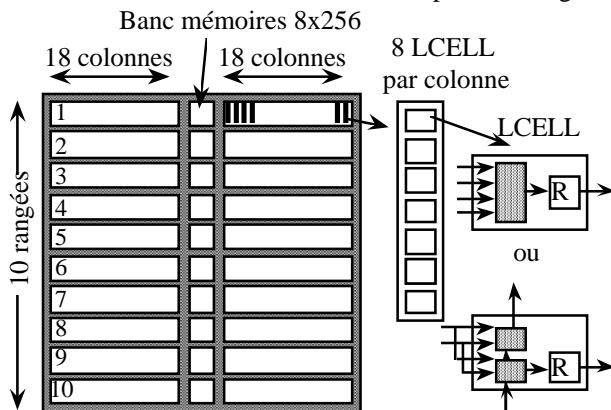


Figure 3 : Architecture du FLEX 10K50

Une LCELL est constituée de logique combinatoire et d'un registre pouvant ou non être utilisé. La logique combinatoire d'une LCELL est programmable dans différents modes, les deux principaux étant le mode "logique

combinatoire" (fonction logique quelconque à 4 entrées) et le mode "arithmétique" (deux fonctions logiques à 3 entrées) qui utilise des signaux à propagations rapides entre LCELL voisines pour réaliser efficacement des additionneurs à propagation de retenue. Entre les LCELL, un réseau de routage permet de propager les signaux. Il faut noter que le temps de propagation d'un signal entre deux LCELL dépend de leur distance et que le passage d'une rangée à une autre est très pénalisant.

4. Multiplieur optimisé

Sur un CLP, la complexité se mesure en nombre de cellules utilisées ce qui est différent de la notion de complexité habituelle (nombre de portes) car une fonction combinatoire peut se décomposer plus ou moins efficacement sur une telle structure.

La multiplication étant une des opérations les plus utilisées dans le circuit, il est nécessaire d'optimiser le bloc multiplieur en nombre de cellules et en performances. Considérant un multiplieur $A \times B$ où A est signé sur M bits et B est signé sur N bits, Une architecture à base de produits partiels nécessite $M \times N$ portes ET suivi d'un arbre d'addition de N mots de M bits. Dans un CLP, les produits partiels sont implémentés dans une seule LCELL ce qui rend cette architecture peu optimale. Une architecture de type BOOTH [3] a un arbre d'addition divisé par 2 en complexité mais nécessite un multiplexeur préalable pour le choix de $(-2A, -A, 0, A, 2A)$ pour les entrées de l'arbre. Le multiplexeur nécessitant 3 LCELL, une des principales optimisations a été de remplacer le codage classique par un codage de BOOTH restreint permettant d'éliminer le cas $2A$ à l'entrée du multiplexeur. Comme $2A = 4A - 2A$, la combinaison $2A$ (groupe de bits de $B=011$) est remplacée par $-2A$ (groupe de bits de $B=100$) pour le poids courant et du même coup ajoute A au groupe de puissance paire supérieure comme l'explique la figure 4 :

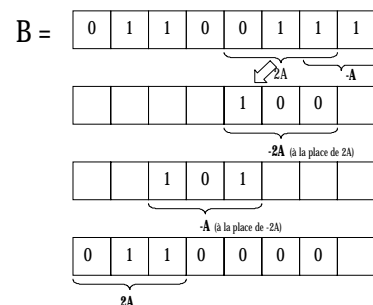


Figure 4 : principe du codage de BOOTH restreint

2 LCELL sont nécessaires car il faut 6 entrées ($A, -2A, -A, 0$ et 2 entrées de sélection) pour réaliser le multiplexeur. En modifiant l'arbre d'addition, le multiplexeur peut se satisfaire de sortir A et $2A$ à la place de $-A$ et $-2A$. Dans ce cas une seule LCELL suffit car les entrées A et $2A$ suffisent à générer $(A, \bar{A}, 2\bar{A}, 0)$. Comme $-A = \bar{A} + 1$, il suffit d'additionner 1 (dit "bit d'inversion") au niveau de l'arbre d'addition.

L'arbre d'addition a un additionneur supplémentaire pour prendre en compte le mot des "bits d'inversion" et des bits de signe comme le montre la figure 5.

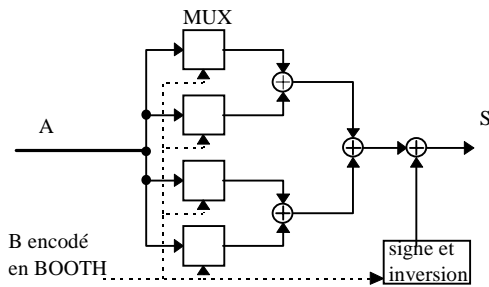


Figure 5 : architecture d'un multiplieur mx8

Si un opérande représente une variable dont les valeurs sont connues à l'avance, l'étape d'encodage de BOOTH restreint peut être supprimée et la variable arrive préencodée sur le multiplieur comme c'est le cas de la variable B de la figure 5.

En utilisant correctement les registres intégrés dans les LCELL et comme l'architecture est de type "pipeline" il est possible d'utiliser une horloge de 40 MHz pour toutes les opérations sans augmentation notable du nombre de LCELL.

Cette architecture a permis de réduire d'au moins 40% la complexité par rapport aux multiplieurs proposés par ALTERA.

5 Choix du banc de filtre

Il existe différents algorithmes permettant d'effectuer de façon efficace un banc de filtres, par exemple par un arbre de filtres demi-bandes [2] ou bien encore par un filtre polyphase [2]. Ce dernier a été choisi car il se partitionne naturellement sur les différentes rangées du CLP. Le principe du filtre polyphase consiste à factoriser les calculs des 16 filtres passe-bande par un unique filtre suivi d'une Transformée de Fourier Rapide (TFR). Dans notre application, le filtre initial contient 32 coefficients réels et la TFR est effectuée sur 16 points.

Afin de réduire la puissance de calcul, un sous-échantillonnage d'un facteur huit est effectué avant le filtrage. Ce sous-échantillonnage n'engendre pas de perte d'information car la largeur de bande du signal en sortie de chaque filtre est divisée par 16. Il réduit le nombre de calcul à effectuer pour chaque nouvel échantillon (f_{max}) à quatre multiplications Réel x Complexe.

Les deux rangées supérieures de LCELL du FLEX sont dédiées à ce filtre. Les deux bancs mémoires permettent d'obtenir 3 lectures et une écriture par échantillon reçu et les LCELL des deux rangées sont suffisantes pour réaliser deux multiplieurs réel par réel (à 40 MHz) ainsi que la logique de contrôle.

La TFR-16 est effectuée en deux passes de radix-4. Chaque passe s'implémente naturellement sur deux rangées : les deux mémoires permettent d'effectuer le réordonnement des données entre deux passes. Les mémoires embarquées du FLEX ayant huit bits de large, un arrondi et une saturation sont effectués à la sortie de chaque radix.

Les quatre rangées restantes sont dédiées respectivement:

- au calcul de l'énergie des échantillons et à leur accumulation dans chaque bande sur une durée 1/8 de bit pour constituer un Méta-échantillon (Mech).

- au calcul de la valeur du bit décodé à partir de chaque Mech calculé.

- les deux derniers bancs mémoires sont dédiés à la recherche de la séquence de synchronisation (voir figure 6.b et 6.c).

Pour la recherche de synchronisation, seules deux mémoires sont disponibles, soit en tout 2 fois 8x256 bits. Le système comportant 8 phases et 16 bandes, il y a donc $8 \times 16 = 128$ recherches de synchronisation qui s'effectuent en parallèle soit 32 bits associés à chaque recherche. Ce nombre est suffisant pour stocker les derniers bits décodés sur la longueur de la séquence d'apprentissage afin de détecter la synchronisation (méthode "a posteriori") mais insuffisant pour obtenir une indication de fiabilité permettant ensuite de choisir la bonne phase et la bonne bande en cas de détections multiples. Il a donc fallu repenser le problème afin d'obtenir un algorithme donnant une indication de fiabilité compatible avec les ressources matérielles restantes.

6 Synchronisation "a priori"

Nous avons utilisé une méthode duale (méthode "a priori") : au lieu de regarder dans le passé les L derniers bits et de les comparer à la séquence d'apprentissage, le système tente de "s'accrocher" à la séquence au fur et à mesure que les bits arrivent. Pour cela, nous utilisons une propriété particulière, nommée P, de la séquence d'apprentissage utilisée, à savoir que la distance de Hamming sur les neuf premiers bits de la séquence d'apprentissage et une version décalée de celle-ci est toujours strictement supérieure à 2. En ne tolérant qu'une seule erreur sur la séquence d'apprentissage, il est possible d'obtenir des simplifications. En effet, une situation où par exemple, les 15 derniers bits décodés $[b_{t-14} \dots b_t]$ correspondent (à une erreur près) aux 15 premiers bits de la séquence d'apprentissage et les 9 derniers bits décodés $[b_{t-8} \dots b_t]$ correspondent (toujours à une erreur près) eux aussi aux 9 premiers bits de la séquence d'apprentissage est impossible car en contradiction avec la propriété P. L'algorithme "a priori" consiste donc à mémoriser en permanence les neuf derniers bits décodés. Dès qu'ils sont à une distance inférieure à 1 des neuf premiers bits de la séquence d'apprentissage, le système "s'accroche" et commence à calculer la fiabilité de la synchronisation (somme des fiabilités des bits). A chaque nouveau bit reçu, le système vérifie que sa valeur est conforme à celle attendue. En cas de différence, s'il y a déjà eu une erreur, le système "décroche" et se remet en mode d'attente.

L'avantage de cet algorithme est que l'état de la poursuite d'une accroche peut se coder sur très peu de bits. Pour chaque phase, il faut :

- 1 bit pour le nombre d'erreurs reçues ;

- 5 bits pour poursuivre l'accroche de 9 à L ;

8 bits pour mémoriser les 8 derniers bits reçus (le neuvième bit étant le bit juste reçu).

Il reste $32 - 14 = 18$ bits pour le calcul de la valeur de la fiabilité ce qui est suffisant pour supprimer les accroches dues au bruit et discriminer correctement la bonne accroche. Son inconvénient est que les fiabilités des huit premiers bits sont omises pour le calcul de la fiabilité de la séquence d'apprentissage.

7 Conclusion

Le système de réception complet a été testé avec succès. Celui ci inclut en plus du démodulateur numérique, l'interface RF, le convertisseur Analogique Numérique et un micro-contrôleur pour traiter le code correcteur d'erreur. La fréquence maximale de fonctionnement est de 40 MHz, et le taux d'utilisation des portes de 60 %. La puissance de calcul du circuit est de 160 Méga-multiplications et 560 Méga-

additions par seconde. Le temps de développement total, incluant l'étude de précision, est de 8 hommes-mois.

Cette implantation a donc validé l'approche CLP pour un circuit de traitement du signal complexe qui aurait nécessité un temps de développement plus long avec une filière ASIC. Cette conception a par ailleurs montré les limites de l'outil de simulation d'ALTERA [4] pour un circuit de cette taille (temps de simulation très long et obligation de créer des procédures en C pour interpréter les fichiers résultats de la simulation). La prochaine version de ce circuit sera effectuée, toujours sur une cible FLEX, avec une méthodologie ASIC. Le circuit sera d'abord décrit en VHDL comportemental. La conformité "bit à bit" sera alors vérifiée avec une description en C de l'algorithme. Les modèles VHDL comportementaux serviront alors de référence au développement d'un VHDL niveau transfert de registre synthétisable.

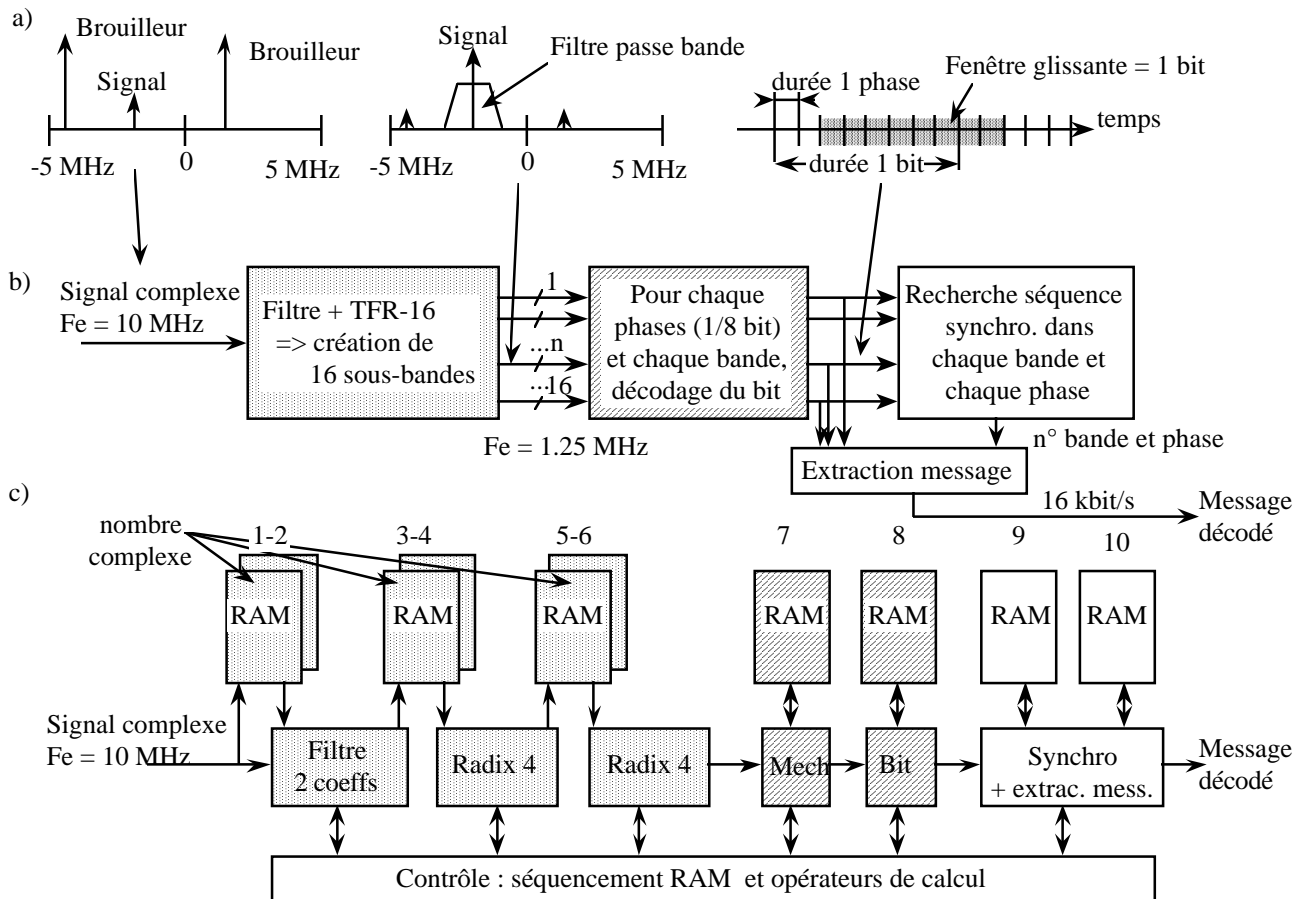


Figure 6 : Représentation de l'algorithme

a) Principe de démodulation

b) Algorithme choisi

c) Décomposition de l'algorithme sur la structure du FLEX10K50

[1] ALTERA, "FLEX10K50 data book",
 [2] Bellanger M. "Traitement Numérique du Signal",
 Collection CNET/ENST

[3] Koren I, "Computer Arithmetic Algorithms", Prentice Hall 1994
 [4] ALTERA MAX+PLUS®II Programmable Logic Development System