Utilisation de la reconfiguration dynamique partielle dans l'implémentation d'un décodeur MIMO V-BLAST

Hongzhi Wang, Pierre Leray, Jacques Palicot et Jean-Philippe Delahaye [1]

[1]IETR/Supélec, Campus de Rennes, Avenue de la Boulaie, CS 47601, 35576 Cesson-Sévigné Cedex, France
Tél. 02.99.84.45.00 - Fax. 02.99.84.45.99
Email : {hongzhi.wang, pierre.leray, jacques.palicot, jean-philippe.delahaye}@supelec.fr

Résumé – Nous proposons dans cet article l'implémentation sur FPGA d'un algorithme de décodage MIMO V-BLAST (Vertical Bell Laboratories Layered Space-Time) Square Root en utilisant la reconfiguration dynamique. L'architecture du décodeur est basée sur l'utilisation d'opérateurs CORDIC (COordinate Rotation DIgital Computer) [3]. L'opérateur CORDIC convient bien pour l'implémentation car il s'appuie seulement sur de simples techniques d'additions et de décalages entre vecteurs. Le but de cette implémentation est de concevoir une architecture minimisant la consommation et les ressources matérielles en termes de nombre de portes logiques dans un FPGA. Cette architecture s'adapte aux débits considérés, ainsi qu'au nombre d'antennes. Pour cela, en fonction des performances attendues notamment de débit et de nombre d'antennes, un nombre minimal d'opérateurs CORDIC est implanté dans la partie fixe d'un FPGA. En revanche, les modules d'interconnexions sont placés dans la partie reconfigurable permettant de changer les contextes en cours d'exécution du décodeur MIMO. L'implémentation du décodeur présente de grands avantages en temps de reconfiguration et ressources matérielles en utilisant la reconfiguration dynamique.

Abstract *– This paper is about the implementation of a MIMO VBLAST (Vertical Bell Laboratories Layered Space-Time) square root decoder in a FPGA using dynamic partial reconfiguration. The decoder architecture is based on CORDIC (COordinate Rotation DIgital Computer) Units. The design implementation aims power saving and area efficiency allowing dynamically changing the interconnections between the fixed modules in the reconfigurable modules. This MIMO square root design method shows the configuration time improvement, area efficiency and flexibility of the decoder by using the dynamic partial reconfiguration method.*

1. Introduction

Avec l'intégration de l'Internet et de nouvelles applications multimédia dans les systèmes de communications sans fil, la demande en terme de débit ne cesse d'augmenter. Plusieurs techniques ont été développées pour répondre à ce besoin. La technique MIMO découverte en 1997 par les chercheurs de Bell Labs reste la plus prometteuse, elle peut augmenter d'une manière substantielle l'efficacité spectrale. Cette technique a rencontré beaucoup d'intérêt ces dernières années et a donné lieu à de nombreux travaux. Parmi les difficultés engendrées par cette technique, l'implémentation des algorithmes de démodulation des signaux MIMO est un sujet d'actualité. Afin de pouvoir gérer la multiplicité des standards de communication, le système MIMO doit supporter différents types de modulations et de propagation. Ce sont les raisons pour lesquelles une architecture reconfigurable trouve tout son intérêt dans les systèmes MIMO.

Les architectures reconfigurables telles que les FPGA permettent une grande liberté de conception des architectures de traitements. Les FPGA offrent la possibilité de reconfiguration dynamique. La reconfiguration dynamique est donc étroitement liée à la notion de reconfiguration partielle qui permet de reconfigurer une partie de FPGA en temps réel sans suspendre les opérations dans les autres parties. La reconfiguration partielle permet de réduire le temps de reconfiguration et l'occupation de la mémoire grâce à une diminution importante de la taille des données de configuration (bitstream).

Dans les travaux précédents [2], nous avons proposé une structure de traitement où le nombre d'opérateurs CORDIC s'adapte aux caractéristiques de débit et de nombre d'antennes pour une implémentation optimale. La reconfiguration était réalisée de manière statique. Nous étendons dans cet article ce travail en proposant la reconfiguration dynamique. Nous introduisons dans cet article l'implémentation sur FPGA d'un algorithme de décodage MIMO V-BLAST (Vertical Bell Laboratories Layered Space-Time) Square Root en utilisant la reconfiguration dynamique. Cette nouvelle conception permet de changer dynamiquement les interconnexions entre les opérateurs CORDIC implantés dans la partie fixe d'un FPGA. En revanche, les modules d'interconnexions sont placés dans la partie reconfigurable permettant de changer les contextes en cours d'exécution du décodeur MIMO.

La suite de l'article est structurée de la manière suivante: la section 2 présente brièvement l'algorithme et l'architecture du décodeur MIMO V-BLAST Square Root. La section 3 décrit la reconfiguration dynamique partielle de FPGA. La section suivante détaille l'implémentation du décodeur « V-BLAST Square Root », que nous avons réalisé ; c'est aussi dans cette section que nous présentons les résultats de la synthèse des différentes architectures sur un FPGA. Enfin, nous insistons sur l'aspect dynamique de la reconfiguration dans l'implémentation du décodeur MIMO et les améliorations que nous envisageons dans les futurs travaux.

2. Algorithme et architecture du décodeur MIMO V-BLAST Square Root

L'algorithme V-BLAST Square Root est proposé dans l'article [5]. L'architecture fonctionnelle du décodeur MIMO « V-BLAST square-root » est illustrée par la figure 1. Elle est composée de 6 modules de traitement. Les entrées sont constituées des messages reçus *r* et des valeurs de la matrice de canal *H*. Les trois premiers modules (M_1, M_2, M_3) effectuent la décomposition de la matrice *H* en utilisant des opérateurs CORDIC (voir l'exemple ci-dessous pour les calculs de $P^{1/2}$ et de Q_α dans le module M_1). Ces modules calculent les grandeurs $P^{1/2}$, Q_a, p_i et $q^*_{a,i}$ qui sont complément définis dans [2] et [5]. Le module suivant M_4 détermine l'ordre optimal de décodage et calcule le vecteur w_i (*nulling vector*). Le module M_5 décide du vecteur symbole transmis et le dernier module M_6 réalise l'annulation d'interférences entre les symboles. Les trois derniers modules (M_4, M_5, M_6) sont basés sur les processeurs élémentaires (PE) constitués d'un multiplieur-accumulateur, un soustracteur et un buffer.

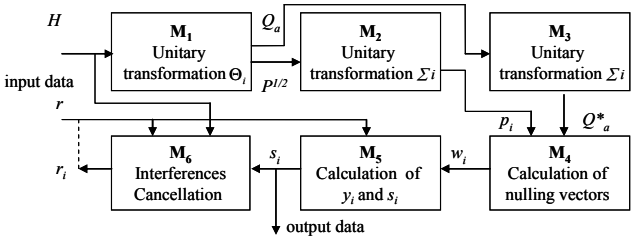

Fig. 1 Architecture fonctionnelle du décodeur MIMO « V-Blast square-root »

Concernant l'implémentation des modules fonctionnels M_1, M_2, M_3, une première solution consiste à instancier autant d'opérateurs CORDIC directement interconnectés que nécessaire pour une structure MIMO données (nombre d'antennes en émission et en réception). La figure 2 illustre une telle implémentation totalement parallèle pour une structure MIMO 2x2 utilisant 29 opérateurs CORDIC. La structure parallèle est conçue pour obtenir un très haut débit, mais elle est coûteuse en ressources matérielles. Pour les applications à plus faible débit, toute la puissance de calcul disponible n'est pas utilisée.

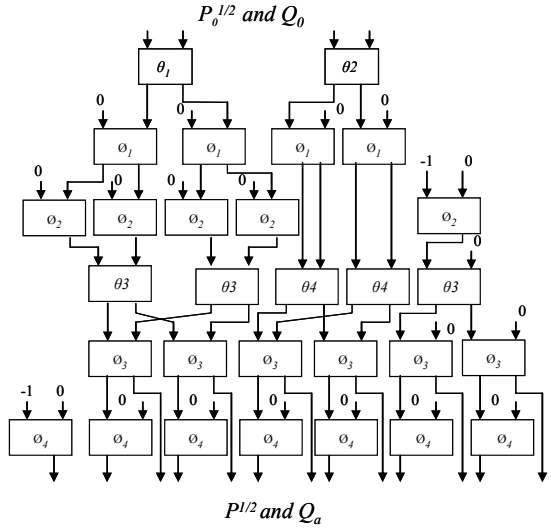

Fig. 2 Structure parallèle de 29 opérateurs CORDIC pour calculer $P^{1/2}$ et Q_a

Pour tendre vers une implémentation optimale du décodeur MIMO, nous avons proposé une architecture reconfigurable dont la structure est fonction des caractéristiques de la transmission. Nous utilisons une structure itérative construite autour d'un certain nombre d'opérateurs CORDIC (par exemple, 3 CORDIC dans la figure 3).

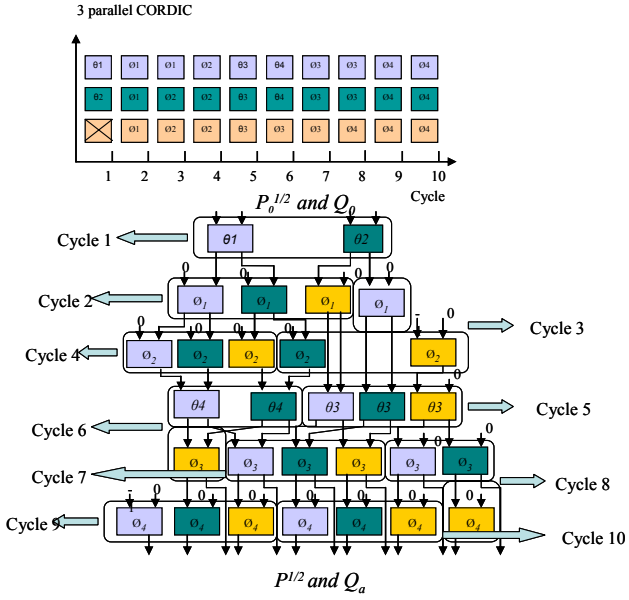

Fig. 3 Organisation et utilisation des différentes structures à base de 3 CORDIC

L'opérateur CORDIC exploite une structure pipeline de 20 étages. L'implémentation statique des interconnexions se fait par l'utilisation d'un grand nombre de multiplexeurs qui changent le contexte d'interconnexions l'un après

l'autre. Les multiplexeurs occupent une surface importante dans un FPGA et consomment beaucoup. En revanche, les multiplexeurs conservent leur état pendant les 20 étapes des opérations CORDIC. Cela nous amène à proposer une implémentation reconfigurable des interconnexions entre les opérateurs CORDIC.

3. Reconfiguration dynamique partielle de FPGA

La reconfiguration dynamique est la capacité d'un système à reconfigurer une partie de ses ressources pendant l'exécution d'une application [4]. La reconfiguration dynamique est donc étroitement liée à la notion de reconfiguration partielle. Et les termes de reconfiguration dynamique et reconfiguration partielle sont le plus souvent associés aux FPGA. Nous considérons que la reconfiguration dynamique d'un FPGA est le multiplexage temporel, d'un ensemble d'éléments reconfigurables. Les éléments configurables de cet ensemble sont réutilisés sous différentes configurations après le lancement d'une application. La reconfiguration dynamique peut être mise en œuvre dans plusieurs cas que nous avons distingués :

1) Reconfiguration pour le multiplexage temporel de blocs de traitement avec une dépendance de données,
2) Reconfiguration dynamique sans dépendance de données.

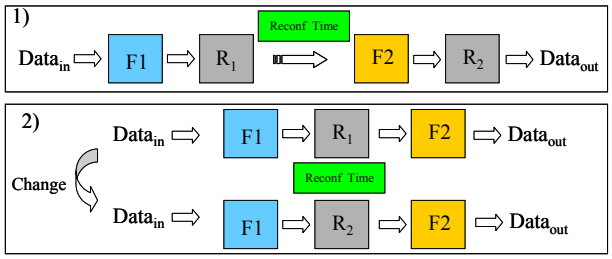

Fig. 4 Différents types de reconfiguration dynamique

La figure 4.1 montre le premier cas de reconfiguration dynamique avec une dépendance de données entre les deux fonctions R_1 et R_2 qui utilisent la même ressource reconfigurable. Dans ce cas F_2 doit attendre la fin de la reconfiguration R_1, comme montré dans la figure 5.1. Il est donc nécessaire d'effectuer une sauvegarde du contexte d'exécution entre chaque reconfiguration, afin de sauvegarder les données en cours de traitement. Dans le deuxième cas (figure 4.2, 5.2), la reconfiguration peut être exécutée à tout moment en dehors de la phase d'exécution du bloc à reconfigurer. Mais il faut aussi assurer que le temps de la reconfiguration ajouté au temps de traitement ne dépasse pas la période d'échantillonnage des données d'entrées dans des applications de type flot de donnée synchrone. Dans cet article, nous utilisons le premier type de reconfiguration et la reconfiguration est exécutée comme une partie du processus du décodeur MIMO.

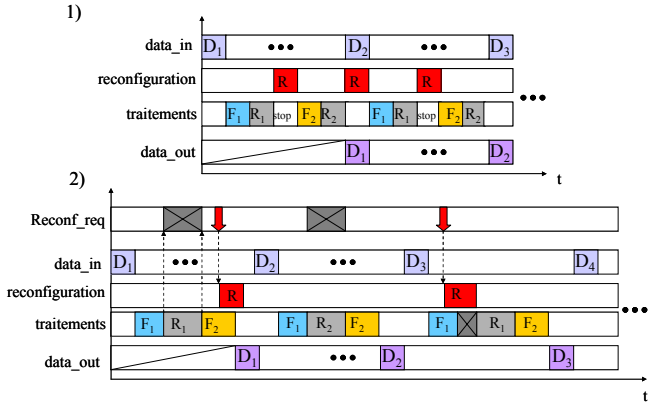

Fig. 5 Représentation temporelle de la reconfiguration

4. Implémentation du décodeur MIMO avec reconfiguration dynamique et partielle

L'architecture globale du décodeur MIMO utilisant la reconfiguration dynamique est illustrée par la figure 6. Elle peut se décomposer en deux parties.

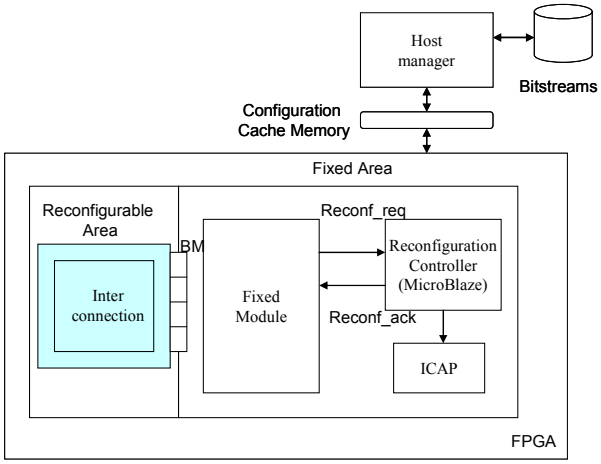

Fig. 6 Architecture du système

La première partie est la partie fixe où les opérateurs CORDIC, les multiplieurs et les additionneurs sont implémentés. La deuxième partie est définie dans des régions reconfigurables d'un FPGA permettant d'accueillir des modules reconfigurables. Dans notre cas, les modules reconfigurables réalisent les interconnexions des opérateurs CORDIC. Les deux parties sont reliées par un Bus Macro (BM) pour assurer que les signaux entre partie fixe et partie reconfigurable seront bien interconnectés après l'opération de reconfiguration partielle.

Les données de reconfiguration sont stockées dans la mémoire du Host. Les chargements des bitstreams sont gérés par le Host. Une première étape d'initialisation pendant laquelle le Host envoie le bitstream total au FPGA permet d'implémenter dans le composant le décodeur MIMO (partie fixe et contexte initial de la partie reconfigurable) ainsi que le contrôleur de reconfiguration constitué notamment d'un module à base de MicroBlaze et de la primitive ICAP (Internal Configuration Port Access). Ensuite les contextes des interconnexions sont changés par les reconfigurations. Lors d'une reconfiguration, seul le bitstream partiel correspondant au contexte du module reconfigurable est rechargé. La taille du bitstream partiel est fortement réduite par rapport à celle du bitstream total permettant de diminuer le temps de reconfiguration. Le séquencement des configurations est prédéfini par le processus du décodeur MIMO.

Ainsi, dans notre approche, les multiplexeurs sont remplacés par des interconnexions câblées reconfigurables, constituées de simples fils de liaisons modifiés à chaque reconfiguration, ce qui permet de diminuer l'encombrement et la consommation. Les signaux de contrôle de configuration de type request/ackowlege (*Reconf_req* et *Reconf_ack* dans la figure 6) permettent de contrôler la configuration du composant et de maintenir l'état des interfaces de transfert de données au cours d'une reconfiguration.

Les modules de l'architecture sont synthétisés individuellement par l'outil de synthèse de Xilinx ISE 6.3. Avec l'aide de l'outil PlanAhead de Xilinx, nous pouvons spécifier les zones de placement de chaque module de l'architecture dans un FPGA Virtex. Puis nous utilisons ISE 6.3 pour effectuer les opérations de placement et de routage de ces modules.

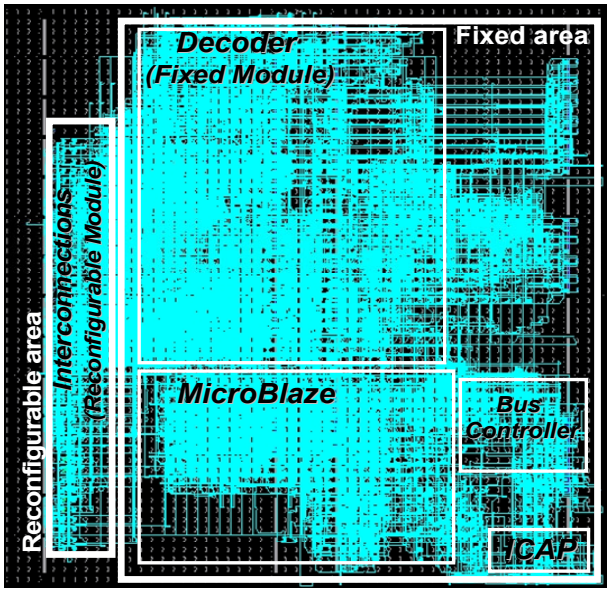

Fig. 7 Implémentation de l'architecture dans un FPGA

La figure 7 montre une vue de l'architecture du décodeur MIMO V-BLAST Square Root après placement-routage dans un FPGA Virtex II de Xilinx. Les résultats obtenus montrent une diminution de 36% des ressources matérielles en termes de nombre de slices dans un FPGA Virtex II de Xilinx. Les résultats de synthèse sont présentés dans le tableau 1.

TAB. 1 : résultats de la synthèse

FPGA Xilinx Virtex-II	Nombre de slices	Nombre de Flip Flops	Temps de reconf
Sans reconfiguration	4505 (40%)	5927 (27%)	16 ms
Partie fixe	2857 (26%)	4766 (22%)	12 ms
Partie reconf 1	85	148	0.4 ms
Partie reconf 2	85	148	0.4 ms

5. Conclusion

Nous avons réalisé l'implémentation du décodeur MIMO basé sur l'algorithme V-BLAST « Square Root » en utilisant la reconfiguration dynamique partielle. Cette architecture offre de grands avantages en terme de flexibilité, de taux d'occupation sur FPGA et de temps de reconfiguration.

Actuellement la performance du décodeur est limitée par le temps de transmission entre le Host et la carte FPGA. Mais cela peut être amélioré par l'implémentation d'un mécanisme de type DMA (Direct Memory Access) dans un FPGA afin d'envoyer directement les bitstreams partiels de la mémoire interne ou externe à l'entrée de la primitive de reconfiguration ICAP de Xilinx. La performance du décodeur peut être encore augmentée en utilisant un FPGA de haute performance Virtex-4 de Xilinx.

Références

[1] G. J. Foshini. *Layered space-time architecture for wireless communication in a fading environment when using multielement antennas*. Bell Labs Technical Journal, pages 41–57, Autumn 1996.

[2] H. Wang, P. Leray, J. Palicot, *Reconfigurable architecture for MIMO systems based on CORDIC operators*, Comptes Rendus Physique, Elsevier, volume 7 septembre 2006, pp 735-750.

[3] Y. H. Hu, *CORDIC-based VLSI Architectures for Digital Signal Processing*," IEEE Signal Processing Magazine, vol. 9, pp. 16, 1992.

[4] J.P. Delahaye, C. Moy, P. Leray, J. Palicot, *Managing dynamic partial reconfiguration on heterogeneous SDR platforms*, SDR Forum 2005, November, Los Angeles, USA.

[5] B. Hassibi, "*An efficient square-root algorithm for BLAST*," http://mars.bell-labs.com/.