

Évaluation comparative de plates-formes reconfigurables et programmables pour les télécommunications de 3^{ème} génération

D. MÉNARD, M. GUITTON, R. DAVID, S. PILLEMENT, O. SENTIEYS

¹LASTI-Université de Rennes I
Équipe de recherche R2D2 (IRISA-INRIA)
6, rue de Kérampont
22300 Lannion FRANCE

Résumé – L'évolution des systèmes de radio-communications cellulaires se traduit par un accroissement important de la complexité des applications à implanter. En particulier, l'utilisation de la technologie WCDMA (*Wideband Code-Division Multiple Access*) dans les systèmes de troisième génération nécessite des capacités de calcul élevées. Dans cet article, les résultats de l'implantation d'un récepteur WCDMA au sein de différentes plates-formes programmables ou reconfigurables sont présentés. Les différentes solutions d'implantation sont comparées en termes de coût, de performances et d'efficacité énergétique. De plus, la méthodologie proposée pour obtenir la spécification en virgule fixe du récepteur est exposée.

Abstract – The cellular radiocommunication third generation systems lead to a significant growth of the application complexity. Especially, the WCDMA (*Wideband Code-Division Multiple Access*) technology requires high-performance computations. The purpose of this paper is to present the results and the analysis of the implementation of WCDMA applications into programmable and reconfigurable platforms. The results of these different implementations are discussed in term of energy efficiency, performances and cost. Moreover, the methodology proposed for the receiver fixed-point specification determination is exposed.

1 Introduction

Les systèmes de radio-communications cellulaires de troisième génération ont été proposés afin d'obtenir un système universel (UMTS : *Universal Mobile Telecommunications System*) permettant de transmettre des données à un débit élevé dans le cadre de services multimédia et liés à l'internet vers un grand nombre d'utilisateurs. Par rapport aux systèmes de seconde génération (GSM, GPRS), ces évolutions se traduisent par un accroissement important de la complexité des applications à implanter. En particulier, l'utilisation des techniques d'accès multiple à répartition par code (WCDMA : *Wideband Code-Division Multiple Access*) nécessite des capacités de calcul élevées.

L'objectif de cet article est de présenter les résultats de l'implantation d'un récepteur WCDMA au sein de différentes plates-formes programmables ou reconfigurables. Les différentes solutions d'implantation sont comparées en termes de coût, de performances et d'efficacité énergétique. De plus, la méthodologie proposée pour obtenir la spécification en virgule fixe du récepteur est exposée.

Les systèmes d'accès multiple à répartition par code étalent le spectre du signal transmis en multipliant les symboles à transmettre par un code (séquence pseudo-aléatoire) de fréquence plus élevée [1]. Cette technique partage une bande de fréquence entre plusieurs utilisateurs possédant des codes différents. En raison des multi-trajets liés aux réflexions présentes au sein du canal de transmission, le signal reçu au niveau du récepteur est composé de plusieurs répliques du signal transmis. La solution utilisée pour améliorer les performances en réception est basée

sur la mise en œuvre d'un *rake receiver* combinant les signaux issus des différents trajets.

Les systèmes de radio-communications cellulaires sont composés d'un terminal mobile et d'une station de base. Les contraintes architecturales pour la conception de ces deux types d'éléments sont de nature relativement différente. Dans le cadre de la station de base, l'architecture doit être flexible afin de supporter les évolutions des standards de radio-communications cellulaires. De plus, le traitement des différents utilisateurs et la mise en œuvre de techniques de réception avancées requièrent des capacités de calcul très élevées. Le mobile étant un produit grand public, son coût doit être le plus faible possible et la contrainte de mobilité impose une consommation d'énergie réduite. Les temps de développement des produits devant être minimisés, la plate-forme utilisée pour implanter le terminal mobile doit être suffisamment flexible pour supporter plusieurs générations de produits [2].

Dans la partie 2, la structure d'un récepteur WCDMA et sa complexité au niveau des traitements sont présentées. Ensuite, la démarche proposée pour déterminer la spécification en virgule fixe du récepteur WCDMA est résumée. Dans la partie 4, les résultats de l'implantation d'un récepteur WCDMA au sein de différentes plates-formes programmables ou reconfigurables sont détaillés.

2 Description du récepteur

Le signal issu de l'antenne est converti en bande de base puis échantillonné et numérisé à une fréquence F_e multiple de

la fréquence F_c du code (3.84 MHz). Le signal numérique complexe est tout d'abord traité par le filtre de réception dont l'objectif est d'annuler l'interférence entre les symboles et de minimiser la puissance du bruit. Ce filtre de réception est composé de deux filtres FIR réels traitant les parties réelles et imaginaires du signal issu du convertisseur analogique/numérique. Les différentes simulations réalisées ont montré qu'un facteur de sur-échantillonnage (F_e / F_c) de 4 et un filtre FIR composé de 64 cellules garantissent des performances correctes en termes de taux d'erreur binaire. Dans ce cas, la complexité du filtre de réception est égale à 2 GMAC/s .

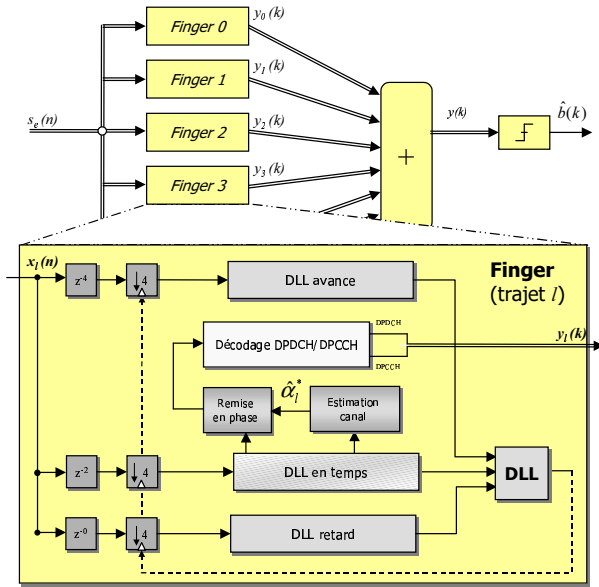


FIG. 1: Synoptique du rake receiver

L'étage suivant, correspondant au *rake receiver* présenté à la figure 1, est composé de différents *fingers* traitant chacun un trajet particulier. L'objectif de chaque *finger* est d'estimer le symbole associé au trajet traité. Pour cela, le signal d'entrée du *rake receiver* est multiplié par le code généré en interne et synchronisé avec le code associé au trajet considéré. Le synoptique des traitements réalisés pour décoder les symboles transmis est présenté à la figure 2. La synchronisation entre le signal reçu et le code généré en interne est réalisée à l'aide d'une boucle d'asservissement (DLL : *delay-locked loop*). Pour combiner de manière optimale les résultats des différents *fingers* en vue d'estimer le symbole transmis, il est nécessaire d'éliminer la phase α_i associée à chaque trajet i . Cette phase est déterminée à l'aide des symboles de référence transmis et connus au niveau du récepteur. La complexité d'un *finger* est de 107 MOPS.

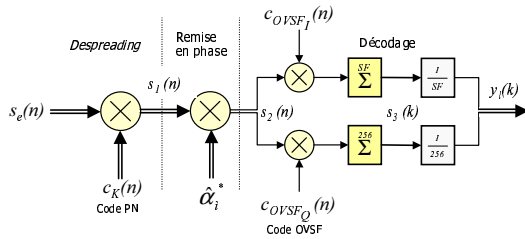


FIG. 2: Synoptique de la partie décodage

3 Détermination de la spécification en virgule fixe

Pour satisfaire les contraintes de coût et de consommation d'énergie associées à cette application, il est nécessaire d'utiliser l'arithmétique virgule fixe. Pour déterminer la spécification en virgule fixe, la méthodologie présentée dans [3] a été utilisée. Elle automatise l'implantation des algorithmes spécifiés en virgule flottante au sein d'architectures en virgule fixe sous contrainte de précision évaluée à travers le Rapport Signal à Bruit de Quantification (RSBQ). L'expression du RSBQ est déterminée automatiquement à l'aide de la méthodologie détaillée dans [4]. Cette approche analytique permet de réduire considérablement le temps d'exécution du processus d'optimisation du format des données par rapport aux méthodes basées sur la simulation de l'application en virgule fixe. En effet, pour ce type d'approche une nouvelle simulation doit être réalisée dès que le format d'une donnée est modifié.

L'utilisation de l'arithmétique virgule fixe va dégrader la précision des calculs et ainsi modifier les performances du récepteur en termes de taux d'erreur binaire (TEB). En conséquence, la valeur minimale du RSBQ ($RSBQ_{min}$) utilisée pour l'implantation est déterminée afin que le TEB obtenu après le passage en virgule fixe reste très proche du TEB en virgule flottante. La valeur $RSBQ_{min}$ est déterminée à l'aide d'une simulation en virgule flottante présentée à la figure 3. L'erreur de quantification globale liée au passage en virgule fixe est modélisée par une source de bruit b_y présente en sortie du système. La puissance de cette source de bruit est augmentée tant que le taux d'erreur binaire n'est pas modifié. La valeur $RSBQ_{min}$ est calculée à partir de la puissance maximale du bruit qui permet de satisfaire la contrainte sur le TEB. Ensuite, la spécification virgule fixe est déterminée afin d'optimiser l'implantation et de satisfaire la contrainte de précision ($RSBQ_{min}$). Cette spécification virgule fixe optimisée est simulée pour vérifier que la contrainte de TEB est réellement satisfaite. Dans le cas contraire, la valeur $RSBQ_{min}$ est ajustée et le processus de conversion en virgule fixe est répété. Cette approche permet de ne réaliser que quelques simulations en virgule fixe. Pour ce récepteur WCDMA, le processus de détermination de la contrainte de précision conduit à une valeur $RSBQ_{min}$ égale à 12.5 dB.

Le signal haute-fréquence issu de l'antenne est converti en bande de base puis numérisé à l'aide d'un convertisseur analogique-numérique (CAN). Un système de contrôle automatique de gain est mis en œuvre afin de maintenir la puissance du signal en entrée du CAN et d'exploiter l'ensemble de la dynamique offerte par le CAN. L'influence du nombre de bits du CAN sur les performances du récepteur a été étudiée dans [1]. La dégradation des performances liée à la quantification du signal est négligeable dès que le nombre de bits utilisés par le CAN est supérieur à 6. En conséquence, la largeur de l'entrée du récepteur WCDMA a été fixée à 8 bits et la spécification en virgule de ce système a été déterminée en fonction de la contrainte de précision ($RSBQ_{min}$). Pour le filtre FIR, la largeur des données en entrée et en sortie du filtre est égale à 8 bits et les résultats de la multiplication et de l'addition sont co-

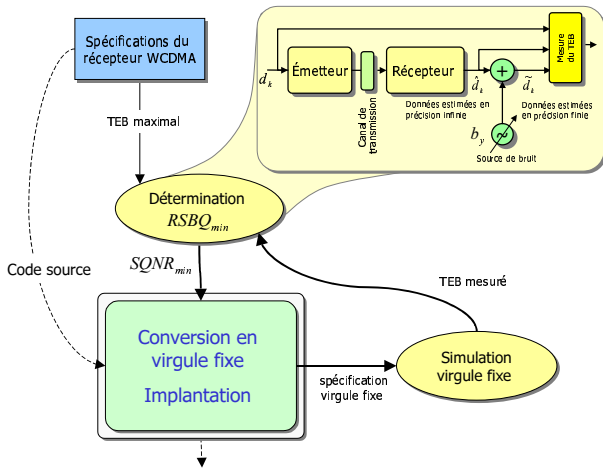


FIG. 3: Méthodologie de conversion en virgule fixe

dés sur 16 bits. Pour le *rake receiver*, la largeur des différentes données pour le décodage DPDCH/DPCCH présenté à la figure 2, est détaillée dans le tableau 1. Excepté pour la première multiplication complexe, les autres opérations sont réalisées en simple précision. Dans le cas d'une multiplication, les largeurs des données en entrée et en sortie de cette opération sont identiques. Ainsi, les bits les moins significatifs de la multiplication sont éliminés.

Données	s_e	s_1	s_2	s_3	y
Largeur (bits)	8	16	16	16	16

TAB. 1: Largeur des données au sein du *rake receiver* présenté à la figure 2

4 Implantations du récepteur

Différentes solutions architecturales peuvent être utilisées pour implanter le récepteur WCDMA. Cependant, l'absence de flexibilité des ASIC nécessite de s'orienter vers des plateformes reconfigurables ou programmables. Les résultats de l'implantation du filtre de réception et du *rake receiver* au sein de l'architecture reconfigurable DART [5, 6], du FPGA Virtex 2000-E [7] et du DSP VLIW C64x [8] sont détaillés. Le *rake receiver* considéré est composé de 6 *fingers* et traite des symboles ayant un facteur d'étalement¹ de 256.

L'architecture DART développée au sein du laboratoire est une architecture reconfigurable au niveau fonctionnel. Deux types d'opérateur sont présents au sein de celle-ci. Les opérations au niveau bit sont réalisées au sein d'un FPGA et les opérations arithmétiques utilisent des chemins de données reconfigurables dynamiquement (DPR). L'architecture DART optimise les interconnexions entre les opérateurs afin d'adapter l'architecture aux motifs de calcul présents dans l'application. Chaque DPR comprend 2 multiplieurs, 2 UAL et 4 mémoires locales disposant chacune de son générateur d'adresses. Ces différents éléments sont reliés entre eux via un réseau d'intercon-

nexions performant et reconfigurable. Les opérateurs arithmétiques supportent un fonctionnement de type SWP (Sub-Word Parallelism) permettant d'exploiter le parallélisme au niveau donnée. La technique SWP consiste à diviser les opérateurs arithmétiques de largeur N afin de pouvoir exécuter en parallèle k opérations sur des fractions de mot de largeur N/k . Au niveau système, l'architecture DART est composée d'une mémoire données, d'une mémoire de configurations et de 4 *clusters* comprenant chacun 6 DPR et un FPGA. La synthèse d'un DPR avec une technologie CMOS $0.18 \mu m$ conduit à une fréquence de fonctionnement de $130 MHz$. Dans le cas de données codées sur 16 bits, chaque *cluster* composé de 6 DPR peut réaliser $1.56 GMAC/s$. Les capacités SWP des opérateurs arithmétiques permettent d'atteindre $3.12 GMAC/s$ pour des données codées sur 8 bits.

L'implantation du filtre de réception au sein d'un *cluster*, conduit à un taux d'utilisation de celui-ci de **63 %**. Pour le *rake receiver*, le taux d'utilisation du *cluster* est de **9.6 %**. À cette puissance de calcul, DART ajoute une excellente efficacité énergétique qui lui permet de se distinguer de ses concurrents. Cette efficacité est principalement obtenue par la minimisation des gaspillages d'énergie inhérents au contrôle de l'architecture et à l'accès aux données. La minimisation du coût énergétique du contrôle de l'architecture est obtenue par la réduction conjointe du volume de données de configuration de l'architecture et de la fréquence des reconfigurations. La hiérarchie mémoire définie pour DART permet dans le même temps de réduire le coût des accès aux données. La flexibilité du réseau d'interconnexions assurant les connexions directes entre les mémoires et les opérateurs, ainsi que la mise en œuvre de chaînes de retard, le coût énergétique des accès aux données est maîtrisé.

Pour le filtre FIR et le *rake receiver*, l'efficacité énergétique moyenne observée atteint **38.8 MOPS/mW**, limitant de ce fait la puissance consommée sur DART à **114.8 mW**. Une étude approfondie de la distribution de la consommation au sein de l'architecture [9] permet de constater l'efficacité énergétique de DART. Sur ce récepteur WCDMA, la part de la consommation induite par le contrôleur de cluster est en effet limitée à moins de 0.8%, lorsque dans le même temps les mémoires et les générateurs d'adresses limite leur consommation à 20 % de l'énergie absorbée par un *cluster*. Cette efficacité énergétique, très supérieure à celle proposée par les architectures concurrentes de DART, s'explique donc par la minimisation des gaspillages d'énergie qui se traduit par une exploitation quasi-optimale des opérateurs. Bien qu'optimisés en consommation, ceux-ci absorbent en effet 79% de l'énergie consommée par le composant.

Les FPGA sont composés d'un ensemble d'éléments logiques reconfigurables. L'implantation matérielle des applications permet d'obtenir des performances élevées en termes de calculs et les possibilités de reconfiguration des FPGA fournissent une flexibilité relativement importante. La famille Virtex inclut un ensemble de blocs logiques reconfigurables (CLB), de mémoire RAM embarquée et d'entrées/sorties configurables. De plus, des opérateurs arithmétiques dédiés tels que des multiplieurs 18×18 bits sont présents. L'implantation dans le Virtex-2000E du filtre de réception correspondant à 2 filtres FIR de 64 cellules conduit à une utilisation de **10 %** des ressources (2188

¹. Rapport entre la fréquence du code F_c et la fréquence des symboles transmis

slices, 1024 bascules). Cette solution a été obtenue avec l'outil *Simplify Pro* de Synplicity et l'outil de placement routage de Xilinx. Le temps d'exécution du filtre est de **38.6 ns** et la puissance estimée avec l'outil XPower de Xilinx est de **919 mW**. L'outil de synthèse de haut niveau BSS a permis de synthétiser l'architecture (chemin de données et contrôleur) nécessaire pour implanter le *rake receiver*. Cette implantation dans le FPGA utilise pour un fonctionnement en temps réel, 300 slices et 20 Kbit de mémoire SRAM embarquée. L'utilisation d'un FPGA plus petit diminuerait la consommation d'énergie mais pour respecter la contrainte temporelle la phase de placement routage serait plus critique.

Métriques	TMS320C64x	
	sans SWP	avec SWP
Temps d'exécution par chip pour un <i>finger</i> (cycles)	12,23	5,45
Temps d'exécution par chip pour un <i>finger</i> (ns)	20,35	9,1
Nombre d'instructions par cycle (IPC) pour un <i>finger</i>	5,3	5,5
Taux d'utilisation du processeur pour le <i>rake receiver</i>	47%	21%

TAB. 2: Résultats d'implantation dans le DSP VLIW C64x (600 MHz)

Le processeur TMS320C64x est un DSP VLIW virgule fixe 32 bits permettant de traiter en parallèle jusqu'à 8 instructions par cycle. Les fréquences d'horloge de ce type de processeur sont comprises entre 300 et 600 MHz. Les capacités SWP du processeur permettent de manipuler des données de largeurs différentes (8, 16, 32, 40, 64 bits). Étant donnée la complexité du filtre de réception, celui-ci ne peut pas être implanté au sein du DSP. Ainsi, uniquement l'implantation du *rake receiver* au sein du DSP a été réalisée. Pour un DSP C64x fonctionnant à 600 MHz, le temps nécessaire pour décoder un symbole à l'aide du *rake receiver* est de **31.3 μ s**. Ainsi, l'exécution en temps réel de cette application nécessite d'utiliser le processeur DSP pendant **47 %** du temps et d'après les résultats présentés dans [10], la puissance consommée est de l'ordre de **1.5 W**. La spécification en virgule fixe optimisée permet d'utiliser les instructions SWP proposées par le processeur. En conséquence, le code intégrant ces instructions SWP conduit à un taux d'utilisation du processeur de **21 %**. Pour évaluer la qualité du code généré par le compilateur C, le nombre d'instructions exécutées par cycle (IPC) a été déterminé. Les techniques de pipeline logiciel, utilisées par le compilateur C, exploitent au sein des boucles le parallélisme proposé par le processeur. Ces techniques conduisent à un code assembleur de bonne qualité possédant, pour cette application, un IPC moyen de l'ordre de **5,5**. Ainsi, les limites de ce type de plates-formes programmables sont plutôt liées aux performances offertes par l'architecture en termes de capacités de calcul qu'aux outils de développement.

5 Conclusion

L'analyse des résultats met clairement en évidence les avantages et les inconvénients de chacune des plates-formes. Les DSP récents basés sur une architecture de type VLIW offrent la flexibilité requise par les systèmes de communications mobiles de troisième génération mais ils permettent uniquement d'implanter le *rake receiver*. L'efficacité énergétique de ce type de DSP est par ailleurs trop faible pour les terminaux mobiles. Dans le cas des stations de base, le nombre d'utilisateurs pouvant être traités par DSP n'est pas assez important pour fournir une solution viable en termes de coût. Les FPGA fournissent la flexibilité et les performances requises mais la consommation d'énergie de ces composants est élevée. Les processeurs reconfigurables associent à la fois la flexibilité, une consommation d'énergie réduite et de hautes performances. Dans ce cadre, l'architecture reconfigurable DART associée à un environnement de développement adapté permet d'implanter efficacement ce type d'application.

Références

- [1] T. Ojanperä and R. Prasad. *WCDMA : Towards IP mobility and mobile internet*. Artech House Universal Personal Communications Series, 2000.
- [2] J. Rabaey. Silicon platforms for the next generation wireless systems. what role does reconfigurable hardware play? In *10th International Conference on Field Programmable Logic and Applications*, Villach, Austria, August 2000.
- [3] D. Menard, D. Chillet, F. Charot, and O. Sentieys. Automatic Floating-point to Fixed-point Conversion for DSP Code Generation. In *International Conference on Compilers, Architectures and Synthesis for Embedded Systems 2002 (CASES 2002)*, Grenoble, October 2002.
- [4] D. Menard and O. Sentieys. Automatic evaluation of the accuracy of fixed-point algorithms. In *IEEE/ACM Design, Automation and Test in Europe (DATE-02)*, Paris, March 2002.
- [5] R. David, D. Chillet, S. Pillement, and O. Sentieys. *SOC Design Methodologies*, chapter A Dynamically Reconfigurable Architecture for Low-Power Multimedia Terminals, pages 51–62. Kluwer Academic Publishers, 2002.
- [6] R. David, D. Chillet, S. Pillement, and O. Sentieys. Mapping future generation mobile telecommunication applications on a dynamically reconfigurable architecture. In *IEEE International Conference on Acoustic Speech, and Signal Processing ICASSP 2002*, May 2002.
- [7] Xilinx. *Virtex-E 1.8V FPGA Detailed Functional Description*. Xilinx, November 2002.
- [8] Texas Instruments. *TMS320C64x Technical Overview*. Texas Instruments, February 2000.
- [9] R. David. *Architecture reconfigurable dynamiquement pour applications mobiles*. PhD thesis, Université de Rennes 1, July 2003.
- [10] Texas Instruments. TMS320C6414/15/16 Power Consumption Summary. Application Report spra811a, Texas Instruments, March 2002.