Analyse et implantation sur FPGA de quelques algorithmes d'égalisation aveugle

Martin Blouin, étudiant 2e cycle

Dr Paul Fortier, directeur de recherche

Abstract: Recently, fractionally-spaced equalizers have allowed a reduction in complexity/cost for CMA (constant modulus algorithm). This project consists in implementing some of these algorithms in FPGA. In order to be sure that the model implemented in FPGA complies with the theoretical results, we realize a stochastic verification. Afterwards, we compare the results of different implementations among them, as well as the equalization quality of each algorithm.

Résumé: Depuis l'apparition de l'égalisation à T/2, de nouveaux algorithmes de plus faible complexité que le CMA (constant modulus algorithm) ont été développés. Dans un premier temps, ce projet consiste à implanter sur FPGA quelques-uns de ces algorithmes. Afin de s'assurer que le modèle implanté sur FPGA est conforme aux résultats espérés, nous effectuons une vérification stochastique. Par la suite, nous comparons les résultats des différentes implantations entre elles ainsi que leur capacité d'égalisation.

Introduction

En communication numérique, la distorsion causée par le canal de transmission peut rendre la détection correcte d'un symbole impossible. Dans les communications sans fil, en plus de ne pas connaître le canal, celui-ci peut changer au cours du temps. D'où l'intérêt de l'égalisation, et en particulier de l'égalisation aveugle lorsqu'il n'est pas possible de transmettre une séquence d'apprentissage. Parmi ces algorithmes, le plus populaire est le CMA (*constant modulus algorithm*).

Depuis l'apparition de l'égalisation à *T*/2, de nouveaux algorithmes de plus faible complexité que le CMA ont été développés [1], [2]. Contrairement à ce qui a été publié au sujet du CMA et des autres algorithmes, nous ne cherchons pas à démontrer les capacités d'égalisation des algorithmes, mais nous cherchons à établir des liens existant entre la capacité d'égalisation de chacun des algorithmes avec leurs résultats d'implantation sur FPGA.

Les algorithmes

Le nombre de bits utilisés pour effectuer l'échantillonnage est un paramètre important dans la conception d'un filtre, puisque l'utilisation en trop d'un bit peut avoir de lourdes conséquences sur les caractéristiques de celui-ci. Par exemple, ce bit en trop pourrait entraîner une dimi-

nution importante de la vitesse de fonctionnement du filtre ou même nécessiter l'utilisation d'un FPGA de plus grande taille. De plus, les capacités d'égalisation d'un canal sont aussi affectées par le nombre de bits utilisés lors de l'échantillonnage du signal reçu. La figure 1 nous montre que l'utilisation du SE-CMA [1] (un algorithme simplifié du CMA) avec 11 bits est équivalent à l'utilisation du CMA avec 8 bits. Notez que ces résultats sont obtenus en simulation à l'aide de la chaîne de communication numérique de la figure 2.

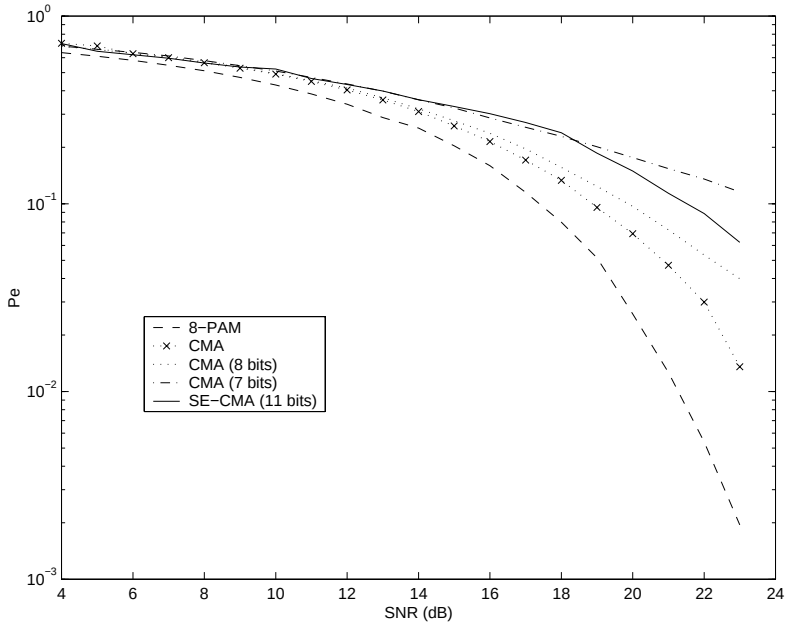

Figure 1 Performances des algorithmes CMA et SE-CMA avec un canal d'ordre 6 et un filtre d'ordre 4.

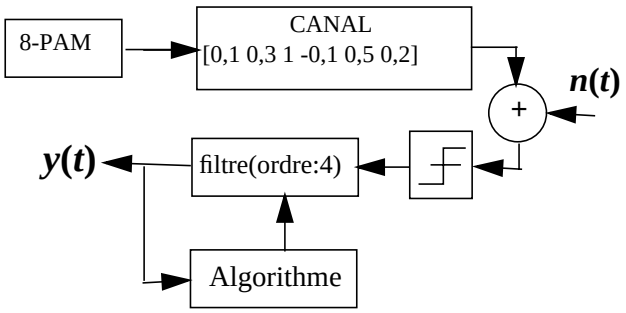

Figure 2 Chaîne de communication numérique utilisée dans l'étude des performances des algorithmes CMA et SE-CMA.

Implantation sur FPGA

Pour implanter ces algorithmes, nous devons effectuer des choix de conception afin de rencontrer certaines spécifications techniques. Puisque nous devons effectuer plusieurs implantations différentes, le filtre doit être très flexible; il doit donc être modulaire et posséder plusieurs paramètres d'implantation. Parmi les paramètres utilisés, l'algorithme, le niveau de parallélisme des calculs effectués, le nombre de bits utilisés pour représenter les symboles, le type de multiplicateur, l'ordre du filtre ainsi que le type de filtre seront les principaux paramètres qui auront le plus d'impact sur les résultats obtenus à la synthèse.

Afin que nos résultats soient crédibles, ces algorithmes sont minutieusement optimisés et utilisent le même noyau.

Vérification

Pour effectuer la vérification d'un circuit, nous avons besoin d'un environnement de vérification que nous appelons communément banc d'essai (*test bench*). Notez qu'un bon environnement de vérification doit nous permettre de vérifier facilement plusieurs fonctionnalités d'un design. Pour vérifier un circuit, une méthode fréquemment utilisée est l'utilisation à des moments prédéterminés de vecteurs contenant les valeurs à appliquer à chacune des entrées de notre circuit ainsi que les valeurs attendues à la sortie. Les valeurs attendues à la sortie sont utilisées par le banc d'essai afin de vérifier si le circuit fonctionne comme prévu.

Un banc d'essai moderne utilise rarement des vecteurs de test, car ceux-ci nécessitent d'être vérifiés manuellement, ce qui rend très difficile la vérification d'un circuit de très grande taille. La vérification stochastique est une solution à ce problème et elle consiste à générer aléatoirement les vecteurs d'entrées du circuit et à vérifier les vecteurs de sorties à l'aide d'un modèle comportemental du circuit [3]. Ce type de vérification peut permettre de réduire le nombre de tests bench développés, donc de réduire le temps de développement.

Pour la vérification du filtre adaptatif réalisé, nous avons décidé d'implanter une version modifiée de la vérification stochastique. Le principe consiste à utiliser, en plus du logiciel de simulation, un logiciel nous permettant de générer les entrées du circuit, de comparer les sorties des deux modèles ainsi que de permettre la création de graphique et de statistique. Dans notre cas, l'utilisation de Matlab nous permettra d'effectuer nos simulations avec un banc d'essai de très haut niveau.

L'utilisation de plus d'un logiciel nécessite l'implantation d'un canal de communication entre ces deux logiciels. Nous avons décidé d'utiliser des fichiers comme méthode de communication car ceux-ci nécessitent de consacrer peu de temps à la réalisation de ce canal (figure 3). Malheureusement, l'utilisation de ce canal engendre des délais supplémentaires à la simulation.

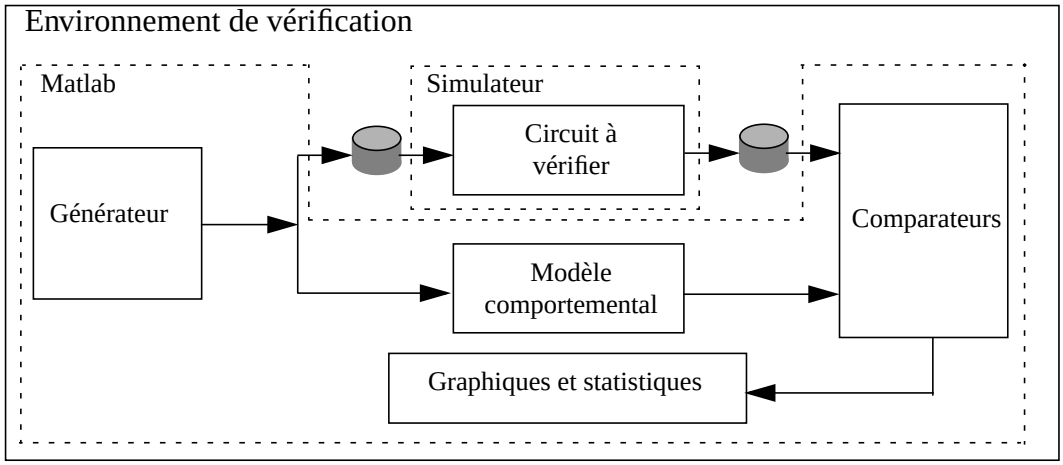

Figure 3 Environnement de simulation.

Conclusion

Les méthodes suggérées pour l'implantation des algorithmes sur FPGA ainsi que pour la vérification nous permettront d'établir les liens existant entre la capacité d'égalisation de chacun des algorithmes et les résultats d'implantation sur FPGA (vitesse, espace, latence interne,...). Dû à la grande flexibilité du filtre adaptatif construit, nous devrions obtenir beaucoup de résultats.

Références

[1] D. R. Brown, P. B. Schniter, C. R. Johnson, "Computationally Efficient Blind Equalization," Thirty-Fifth Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, 1997.
[2] P. Schniter and C. R. Johnson, "The dithered signed-error constant modulus algorithm", ICASSP 1998.
[3] Sandi Habinc, Peter Sinander, "Accelerated Verification of Digital Devices Using VHDL", FDL'98, september 8-11, 1998.