



Solution ASF pour un simulateur matériel du canal de propagation MIMO hétérogène

Bachir Habib, Gheorghe Zaharia, Ghaïs El Zein

► To cite this version:

Bachir Habib, Gheorghe Zaharia, Ghaïs El Zein. Solution ASF pour un simulateur matériel du canal de propagation MIMO hétérogène. Journées Scientifiques URSI 2013, Mar 2013, Paris, France. pp.227-228. hal-00806693

HAL Id: hal-00806693

<https://hal.science/hal-00806693>

Submitted on 2 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Titre : Solution ASF pour un simulateur matériel du canal de propagation MIMO hétérogène

Title: ASF solution for a hardware simulator of MIMO heterogeneous propagation channel

Bachir Habib, Gheorghe Zaharia, Ghaïs El Zein

*IETR-INSa, UMR CNRS 6164, 20 Av. des Buttes de Coësmes, CS 70839, 35708 Rennes Cedex 7, France
bachir.habib@insa-rennes.fr*

Mots clés : Simulateur matériel; canal MIMO; FPGA
Hardware simulator; MIMO channel; FPGA

Introduction

L'architecture du bloc numérique du simulateur matériel est réalisée sur un circuit logique programmable (FPGA) Virtex-IV [1] qui a une grande flexibilité dans l'implémentation d'algorithmes. Les modèles de canaux utilisés par le simulateur peuvent être des modèles de canaux standards, comme les modèles TGn [2], ou des mesures obtenues à l'aide d'un sondeur de canal MIMO conçu à l'IETR [3]. En général, les canaux de propagation radio sont simulés en utilisant un filtre à Réponse Impulsionnelle Finie (FIR) ou par la Transformée de Fourier Rapide (FFT) [4-5]. Dans [6], nous avons montré que l'architecture temporelle est la meilleure en termes d'occupation sur FPGA, de précision des signaux de sortie et de latence. C'est elle qui sera considérée dans cet article. Ce travail s'inscrit dans le cadre du projet PALMYRE II, soutenu par la Région Bretagne. Les principales contributions de cet article sont les suivantes :

- Des tests ont été réalisés à l'intérieur et à l'extérieur des bâtiments [6] pour des environnements fixes. Dans cet article, les tests sont faits pour une succession d'environnements « indoor ».
- Un ASF est mis en œuvre dans l'architecture du simulateur. Une étude approfondie montre que ce facteur augmente le SNR lorsque les signaux d'entrée (x) ou les réponses impulsionnelles (h) sont atténués.

Le reste de ce papier est organisé comme suit. La section 2 présente les caractéristiques du canal. La section 3 décrit la mise en œuvre matérielle de l'architecture du bloc numérique, alors que la section 4 analyse la précision de l'architecture. Enfin, la section 5 donne quelques conclusions.

1. Caractéristiques du canal

Une description détaillée des modèles TGn est présentée dans [2]. La fréquence centrale est $f_c = 5$ GHz et la fréquence d'échantillonnage des signaux est $f_s = 180$ MHz pour une largeur de bande $B = 80$ MHz. Pour obtenir un canal variant dans le temps, des évanouissements de type Rayleigh sont générés [2]. Le scénario proposé est représenté par la Fig. 1. Il contient trois environnements dont les caractéristiques sont données dans le Tableau 1. v est la vitesse du mobile, f_d est la fréquence Doppler, f_{ref} est la fréquence de rafraîchissement des profils MIMO, t est la durée de la simulation, d est la distance parcourue et N_p est le nombre de profils MIMO utilisés.

2. Architecture du bloc numérique

Quatre filtres FIR sont utilisés pour mettre en œuvre un canal MIMO 2x2. L'architecture d'un filtre FIR a été déjà présentée [6]. Dans cet article, le ASF est utilisé. La solution proposée consiste à multiplier x et h par des facteurs k_x , respectivement k_h qui augmentent les signaux de sortie et qui permet de les quantifier sur un grand nombre de bits afin de réduire l'erreur à la sortie. Si $x_{max} = \max(|x|) < 0.25$ V, les signaux d'entrée seront multipliés par 2^{k_x} où k_x est l'unique entier qui vérifie:

$$\frac{1}{2} < 2^{k_x} \cdot x_{max} < 1, \quad \text{alors:} \quad k_x = \left\lceil \log_2\left(\frac{1}{x_{max}}\right) \right\rceil - 1 \quad (1)$$

où $[u]$ est le plus petit entier qui est plus grand que (ou égal à) u . $h_{max} = \max(|h|) < 0.5$, h seront multipliées par 2^{k_h} où:

$$k_h = \left\lceil \log_2\left(\frac{1}{h_{max}}\right) \right\rceil - 1 \quad (2)$$

La troncature brutale, réalisée avant le convertisseur numérique-analogique (CNA), est nécessaire pour quantifier le signal de sortie sur 14 bits (à cause du CNA). Une meilleure solution est la troncature glissante [6] qui sélectionne les bits significatifs de poids fort. Dans ce cas, le facteur de glissement est 2^{k_y} . Ainsi, $ASF = k_x + k_h - k_y$ est envoyé à un

amplificateur à gain contrôlable pour rétablir la vraie valeur des signaux de sortie. Le résumé d'utilisation du Virtex-IV SX35 est donné dans le Tableau 2 avec le circuit utilisé pour recharger automatiquement les coefficients du canal. Pour les trois environnements, le nombre maximum de trajets est 18 pour chaque canal SISO. Pour chaque profil MIMO, les atténuations des trajets sont stockées dans un bloc RAM. Pour passer d'un environnement à un autre et pour être capable de contrôler le changement des retards des trajets, il suffit de connecter une entrée de chaque multiplieur du filtre FIR à la sortie du bloc du registre à décalage correspondant à ce retard.

3. Précision

Pour déterminer la précision des signaux de sortie assurée par cette architecture, une comparaison est faite entre les signaux théoriques et les signaux Xilinx obtenus à la sortie du simulateur. A l'entrée du simulateur on considère des signaux Gaussiens de la forme :

$$x_1(t) = x_2(t) = x_m e^{-\frac{(t-m_x)^2}{2\sigma^2}}, \quad 0 \leq t \leq W_t \quad (3)$$

où : $W_t = 400T_s$, $m_x = 200T_s$, $\sigma = 50T_s$, $x_m = 0.5$ V. Ces paramètres sont choisis pour observer l'effet de h sur les signaux de sortie. L'erreur relative globale et le SNR global sont donnés dans [6]. Fig. 2 présente la variation du SNR global pour tous les profils.

Conclusion

Cet article présente un simulateur matériel du canal de propagation MIMO hétérogène avec différents retards. Le facteur ASF est introduit pour augmenter le SNR des signaux de sortie. L'architecture proposée est implémentée sur un FPGA Virtex-IV pour un canal MIMO 2x2. Les simulations faites en utilisant un FPGA Virtex-VII nous permettront de simuler des canaux MIMO de taille 8x8. Une interface graphique est conçue pour permettre à l'utilisateur de sélectionner les paramètres du canal.

Références bibliographiques

- [1] 'Xilinx: FPGA, CPLD and EPP solutions', www.xilinx.com.
- [2] Erceg, V., Shumacher, L., Kyritsi, P., et al. : 'TGn Channel Models', IEEE 802.11- 03/940r4, May 10, 2004.
- [3] H. Farhat, R. Cosquer, G. Grunfelder, L. Le Coq and G. El Zein, 'A dual band MIMO channel sounder at 2.2 and 3.5 GHz', IMTC, Victoria, BC, Canada, May 2008.
- [4] S. Picol, G. Zaharia, D. Houzet and G. El Zein, 'Features of the Digital Block of a Hardware Simulator for MIMO Radio Channels', IEEE ISSCS, Iasi, Romania, July 2007.
- [5] H. Eslami, S.V. Tran and A.M. Eltawil, 'Design and implementation of a scalable channel Emulator for wideband MIMO systems', IEEE Trans. on Veh. Technol., vol. 58, no. 9, pp. 4698-4708, Nov. 2009.
- [6] B. Habib, G. Zaharia and G. El Zein, 'Digital Block Design of MIMO Hardware Simulator for LTE Applications', IEEE ICC, Ottawa, Canada, 2012.

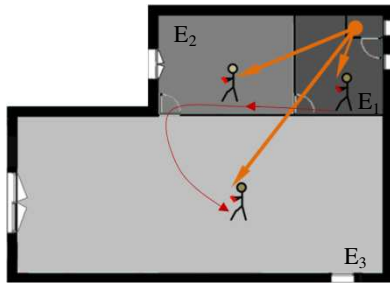


Figure 1. Scénario étudié

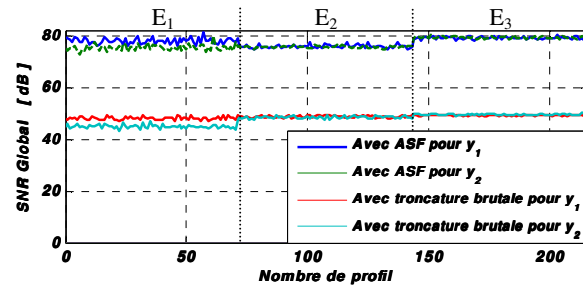


Figure 2. Variation du SNR global

Tableau 1. Caractéristiques des environnements

	Modèle	v (km/h)	f_d (Hz)	f_{ref} (Hz)	t (s)	d (m)	N_p
E ₁	TGn B	3	14	30	2,4	2	72
E ₂	TGn C	3	14	30	4,8	4	144
E ₃	TGn E	3	14	30	7,2	6	216

Tableau 2. Occupation sur Virtex-IV

	Utilisé	Disponible	Utilisation
Bascules	2096	30720	7 %
Slices occupées	2681	15360	18 %
Buffers IO	40	448	8 %
FIFO16/RAMB16s	164	192	86 %
DSP48s	72	192	38 %