

Emulation de Systèmes Cyber-Physiques sur FPGA

Maelic LOUART^{1,2}, Jean-Christophe LE LANN¹, Frédéric LE ROY¹, Abdel BOUDRAA² Jean-Jacques SZKOLNIK²

¹Lab-STICC, UMR CNRS 6285, ENSTA Bretagne, 2 Rue François Verny, 29200 Brest, France

²IRENav,EA3634, BCRM Brest CC 600, 29240 BREST Cedex 9, France

(maelic.louart, jj.szkolnik, abdel.boudraa)@ecole-navale.fr,

(jean-christophe.le-lann, frederic.le-roy)@ensta-bretagne.fr

Résumé – Les FPGAs permettent aujourd’hui la conception de systèmes embarqués complexes complets capables de couvrir une gamme extrêmement large d’applications, allant des systèmes embarqués classiques à la virtualisation dans le cloud. Toutefois, le potentiel des FPGAs ne s’arrête pas là. Leur flexibilité et leur parallélisme naturel invitent à explorer leur utilisation dans des domaines plus confidentiels. Dans cet article, nous nous intéressons à l’émulation de systèmes cyber-physiques complets sur FPGA. Notre cas d’étude présente l’émulation d’un système constitué d’une multitude de navires communiquant selon le protocole d’identification AIS. L’étude démontre un facteur d’accélération de $\times 654$ par rapport à l’exécution purement logicielle, et permet d’esquisser un flot de conception adapté au domaine de l’émulation de tels systèmes sur FPGAs.

Abstract – FPGAs today enable the design of complete complex embedded systems capable of covering an extremely wide range of applications, from classical embedded systems to cloud virtualization. However, the potential of FPGAs does not end there. Their flexibility and natural parallelism invite to explore their use in more confidential domains. In this paper, we focus on the emulation of complete cyber-physical systems on FPGAs. Our case study presents the emulation of a system consisting of a multitude of ships communicating according to the AIS identification protocol. The study demonstrates a speed-up factor of $\times 654$ compared to pure software execution, and allows us to outline a design flow adapted to the field of emulation of such systems on FPGAs.

1 Introduction

Les FPGAs possèdent plusieurs vertus remarquables [1] qui leur confèrent un statut particulier dans le domaine de l’Électronique. Ils permettent désormais de concevoir des systèmes embarqués complexes complets et sont en particulier attrayants pour de faibles volumes de production. On peut les trouver dans des systèmes embarqués variés, allant du multimedia aux robots interplanétaires, du médical au militaire, etc. Ici, les capacités de reconfiguration dynamique, de versatilité, de synchronisation et d’accélération des calculs ainsi que la faible consommation d’énergie font des FPGAs des plateformes haute performance, désormais incontournables [2].

Toutefois, dès les années 2010, des chercheurs ont proposé d’utiliser les FPGAs hors des sentiers battus. En particulier, Givargis et Vahid ont proposé d’utiliser des FPGAs comme émulateurs de tout autres systèmes : il s’agissait d’émuler le fonctionnement d’un poumon artificiel [3]. Le rôle du FPGA était alors de permettre le test et la mise au point de ventilateurs externes préexistants : le FPGA jouait le rôle d’un organe artificiel, modélisé par un système d’équations différentielles établi avec des spécialistes de la respiration. Il devenait facile de placer ce poumon artificiel dans des situations complexes ou difficiles à gérer pour le ventilateur. De tels ventilateurs rentrent dans la catégorie des systèmes cyber-physiques (CPS) : ils sont eux-mêmes constitués de microcontrôleurs, de capteurs et d’ac-

tionneurs, et sont bien évidemment en interaction forte avec leur environnement (ici le poumon). Ces travaux remarquables n’ont pas selon nous reçu une attention suffisante.

Nous proposons, ici, une méthode alternative, mais en ligne avec les ambitions de ces auteurs, dans un tout autre cas d’application : il s’agit pour nous de mettre au point un récepteur de communication maritime. De manière similaire, ce récepteur est un assemblage complexe de blocs de traitement du signal, et est également sensible à son environnement physique. Cet environnement physique est simulé et correspond à un canal de communication hertzien, soumis aux lois physiques caractéristiques d’un milieu maritime. Notre manière de procéder est toutefois très différente de celles de Givargis, Vahid et Miller : en premier lieu, nous disposons, aujourd’hui, de FPGAs contenant une quantité de ressources largement supérieure à celle dont bénéficiaient les auteurs à la date de publication. En second lieu, nous allons démontrer l’utilisation possible de la synthèse comportementale (HLS) afin de considérer des modélisations à très grande échelle (celle de plusieurs navires émettant des trames AIS simultanément, voir section 3). Enfin, notre méthode permet, à la fois de simuler cet environnement physique, mais aussi d’émuler différents systèmes embarqués communicants sur un même FPGA, et qui sont des prototypes viables, utilisables dans un environnement réel. Il s’agit selon nous des premiers travaux qui poussent aussi loin l’utilisation des FPGA à des fins d’émulation et de simulation.

2 Méthode d'émulation de systèmes cyber-physiques sur FPGA

Comme exprimé dans l'introduction, nous proposons une nouvelle méthode de simulation de CPSs qui, à partir d'un modèle exprimé en C++, utilise la HLS pour générer le simulateur complet exécutable sur un FPGA. L'intérêt d'utiliser du code C++ est la disponibilité quasi-systématique de codes C/C++ hautement spécifiques à un domaine, capables de simuler des parties du comportement de presque tous les CPS (phénomènes physiques, mécaniques, chimiques, etc.). Notre méthode divise le CPS en composants : un d'entre eux simule l'environnement et les autres émulent des systèmes embarqués. La description des comportements des composants se fait en C++ et l'interaction entre composants est programmée par l'utilisation de la directive *dataflow* fournie par la HLS. Cette directive implémente automatiquement des canaux sous forme de FIFO ou PIPO entre composants. Du fait de l'utilisation de cette directive, l'architecture du CPS simulé est limitée à un graphe orienté acyclique.

De plus, un composant *Pilotage* est ajouté sur le FPGA pour permettre une interaction classique, depuis le PC hôte, avec le simulateur : commandes de démarrage, d'arrêt, modification de l'environnement simulé. Et un composant *Surveillance* est ajouté pour permettre d'observer la dynamique de certains signaux simulés sur le FPGA.

3 Cas d'application de notre méthode de simulation

3.1 Le système d'identification automatique

Nous appliquons notre méthode pour simuler un ensemble de navires qui se déplacent et échangent des messages radio en s'appuyant sur le système d'identification automatique (AIS). Ce système améliore la sécurité du trafic maritime en permettant l'échange automatique d'informations de navigation entre les navires (informations dynamiques (position, vitesse, route, cap) et informations statiques (identité, port de départ, etc.)). Les informations sont échangées sur deux fréquences, 161,975 MHz et 162,025 MHz, appartenant à la bande des très hautes fréquences (VHF). Depuis 2002, ce système est obligatoire pour les navires internationaux de plus de 300 tonnes de jauge brute naviguant dans les eaux internationales et pour les navires à passagers quelque soit leur taille.

3.2 Besoin d'émulation

Pour développer un transpondeur AIS fiable, son comportement doit être testé dans des environnements variés reproduisant de manière complète toutes les conditions d'utilisation possibles du transpondeur. Cependant, obtenir un tel flux de tests, à partir de mesures récupérées dans un environnement réel, reste difficile. C'est pourquoi, pour résoudre ce problème, nous avons créé un cadre de simulation pouvant générer une grande

variété de scénarios *synthétiques* contenant des messages et des signaux AIS. Les signaux générés peuvent reproduire les caractéristiques matérielles des transpondeurs (erreurs sur les fréquences des porteuses) et évoluer en fonction des conditions environnementales (distance entre l'émetteur et le récepteur, SNR). De plus, le simulateur peut émuler plusieurs transpondeurs AIS en même temps, pour reproduire un espace maritime chargé en bateaux. La simulation fonctionne entièrement sur FPGA et est synthétisée à l'aide de la HLS. Un exemple de l'architecture du simulateur est présenté sur la FIGURE 1 où trois transpondeurs communiquent.

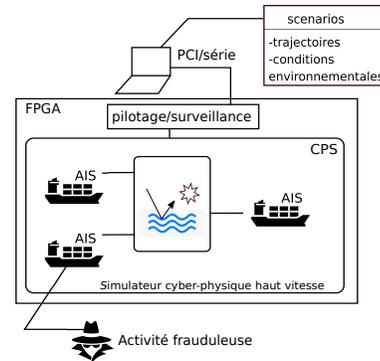


FIGURE 1 – Notre proposition : l'utilisation d'un FPGA comme plateforme de simulation de systèmes cyber-physiques.

3.3 Modèle du CPS simulé

L'architecture détaillée du simulateur de la FIGURE 1 est présentée sur la FIGURE 2. Sur cette figure, trois types de composants sont affichés et ont été modélisés. Un *Émetteur* qui agit comme un transpondeur AIS qui, à partir d'une trame NMEA, envoie un signal AIS modulé en bande de base. Un *Récepteur* qui agit comme un transpondeur AIS, qui reçoit uniquement les signaux AIS, en bande de base, et les démodule pour en extraire la trame NMEA. Enfin, un *Canal* qui simule l'environnement pour appliquer certaines lois physiques aux signaux AIS échangés (bruit et affaiblissement de propagation) et reproduit certaines imperfections matérielles des *Émetteur* et *Récepteur* (différence fréquentielle entre leurs fréquences porteuses de modulation). L'*Émetteur* et le *Récepteur* représentent les systèmes embarqués du CPS et le *Canal* représente l'environnement.

Nous représentons également sur la FIGURE 2 deux composants spécifiques, un composant *Pilotage* et un composant *Surveillance*. Le composant *Pilotage* pilote l'environnement maritime simulé, gère, à partir d'un réglage initial effectué à l'initialisation du simulateur, l'évolution des données dynamiques de chaque navire au cours de la simulation et, avec ces données et les informations statiques, crée les trames NMEA transmises à chaque *Émetteur*. Le composant *Surveillance* permet d'observer la dynamique de certains signaux simulés sur le FPGA.

Notez que l'architecture est modélisée par un graphe orienté acyclique : le composant *Pilotage* est situé au début du graphe

et le composant *Surveillance* à la fin. Aussi, certaines parties du système sont dédiées à la simulation pure, tandis que certains éléments (comme notre *Récepteur*) peuvent être considérés comme un prototype viable d'un futur dispositif embarqué. En effet, le composant *Récepteur* a parfaitement réussi à décoder des signaux AIS réels enregistrés dans la rade de Brest.

3.4 Synthèse matérielle

Nous avons expérimenté notre méthode de conception du simulateur présentée dans la partie 2. L'ensemble du modèle CPS représente environ 4000 lignes de code C++. Des variables à virgule fixes sont utilisées. Les résultats de la synthèse des trois composants sont présentés dans le TABLE 1. Pour les composants *Surveillance* et *Pilotage* rien n'est présenté car les valeurs sont négligeables. La cible de la synthèse était un FPGA Xilinx Ultrascale+ HBM sur une carte VCU128 (environ 10^6 cellules logiques, 3K DSP, etc). L'outil HLS était Vitis HLS.

Le pourcentage d'occupation du FPGA est au maximum de 7% (mémoires BRAM) lorsque chaque composant est implémenté. Ce faible chiffre laisse de l'espace libre permettant de complexifier l'environnement simulé et d'augmenter le nombre de transpondeurs émulsés en même temps. Ainsi, nous montrons dans le TABLE 2 de la partie 4.2, qu'au maximum, avec les composants décrits dans la partie 3.3, 22 *Émetteurs* peuvent être simulés en même temps sur le FPGA.

4 Simulations de CPS

4.1 Présentation des scénarios et environnements simulés

Pour illustrer la capacité de notre simulateur à simuler plusieurs navires à la fois, trois scénarios ont été simulés. Un contenant un seul navire, un autre 5 et un autre 10. L'ajout d'un navire correspond à l'ajout d'un composant *Émetteur* sur le FPGA. En plus de ces composants *Emetteur*, un composant *Canal*, *Récepteur*, *Pilotage* et *Surveillance* sont ajoutés. Pour chaque scénario, la simulation dure jusqu'à ce que le *Récepteur* reçoive 1000 messages AIS. Les messages sont transmis successivement par chacun des *Émetteurs*. Pendant qu'un des *Émetteurs* transmet un signal contenant un message AIS modulé, les autres ne transmettent que du bruit.

Par ailleurs, pour montrer l'intérêt de notre composant *Canal*, les performances de notre *Récepteur* ont été évaluées statistiquement pour plusieurs valeurs de SNR et de décalage fréquentielle entre la porteuse de l'*Émetteur* et du *Récepteur*.

TABLE 1 – Ressources utilisées par chaque composant du simulateur.

| | BRAM | DSP | FF | LUT |
|-------------------|---------|--------|-----------|-----------|
| Simulateur | 293(7%) | 66(0%) | 30508(1%) | 41665(3%) |
| Émetteur | 97 | 1 | 8718 | 16357 |
| Canal | 97 | 13 | 1478 | 4312 |
| Récepteur | 99 | 52 | 19904 | 20754 |

4.2 Résultats de la simulation

Nous comparons, dans le TABLE 2, le temps de la simulation logicielle, au temps de la simulation matérielle, pour les trois scénarios présentés ci-dessus. La simulation logicielle se fait sur un processeur Intel Core I5 standard à 1.7GHz avec 16Gb de RAM et la simulation matérielle sur le FPGA présenté ci-dessus. Dans ce tableau, nous affichons aussi le taux d'occupation du FPGA et nous prédisons les performances que l'on aurait obtenu si nous avions simulé 22 *Émetteurs* en même temps. Pour ce nombre d'*Émetteurs*, le taux d'occupation des BRAM atteint 98%, le FPGA ne peut pas émuler plus d'*Émetteurs* en même temps.

Pour la simulation matérielle, le temps d'exécution des trois scénarios est très courts et reste le même alors que le nombre d'*Émetteurs* à émuler augmente. Cela souligne la capacité des FPGAs à paralléliser les calculs et leur scalabilité. Le gain d'accélération est déjà de $\times 183$ avec seulement un *Emetteur*. Ce gain est obtenu sans l'utilisation de directives d'optimisation, sous Vitis HLS, comme le déroulage de boucle ou les pipelines, ce qui laisse encore une marge d'accélération. Lorsque le nombre d'*Emetteur* augmente, le gain d'accélération augmente aussi, car les composants *Emetteur* sont exécutés en parallèle. Ainsi, nous prédisons que le gain d'accélération peut atteindre $\times 654$ pour 22 *Emetteurs*.

L'utilisation de variables à virgule fixe introduit des erreurs, qui influencent la précision des données dynamiques générées. Une analyse statistique sur 1000 valeurs montre que ces erreurs suivent une distribution gaussienne centrée avec un écart-type de $\sigma_{lat} = 0.32m$ pour la latitude et de $\sigma_{lon} = 0.37m$ pour la longitude. Ces écart-types sont négligeables par rapport à l'écart-type d'un GPS (5m), qui est justement le capteur utilisé par l'AIS pour connaître la position des bateaux. Cela valide la capacité de notre simulateur à simuler des environnements maritimes contenant plusieurs bateaux se déplaçant suivant des trajectoires définies précisément.

Pour finir, afin de montrer la capacité du simulateur à simuler différentes conditions environnementales, les performances du démodulateur GMSK, contenu dans le composant *Récepteur*, sont évaluées pour différentes valeurs de SNR et de décalage fréquentielle. Les résultats obtenus sont présentés sur la courbe de la figure FIGURE 3 qui montre l'évolution du taux d'erreur binaire (BER) en fonction du SNR et du décalage en fréquence des porteuses (+0Hz, +200Hz et +400Hz). Les performances obtenues sont similaires à celles d'autres démodulateurs GMSK rencontrés dans la littérature, et permettent de définir les conditions d'utilisation limites de notre composant *Récepteur* qui est un prototype viable.

TABLE 2 – Comparaison du temps de simulation entre processeurs et FPGA

| nb. Émetteurs | Soft. | Hard. | Gain | taux d'occupation |
|---------------|-------|-------|------|-------------------|
| 1 | 576s | 3.14s | 183 | 7% |
| 5 | 856s | 3.14s | 272 | 24% |
| 10 | 1210s | 3.14s | 386 | 46% |
| 22 (predict.) | 2050s | 3.14s | 654 | 98% |

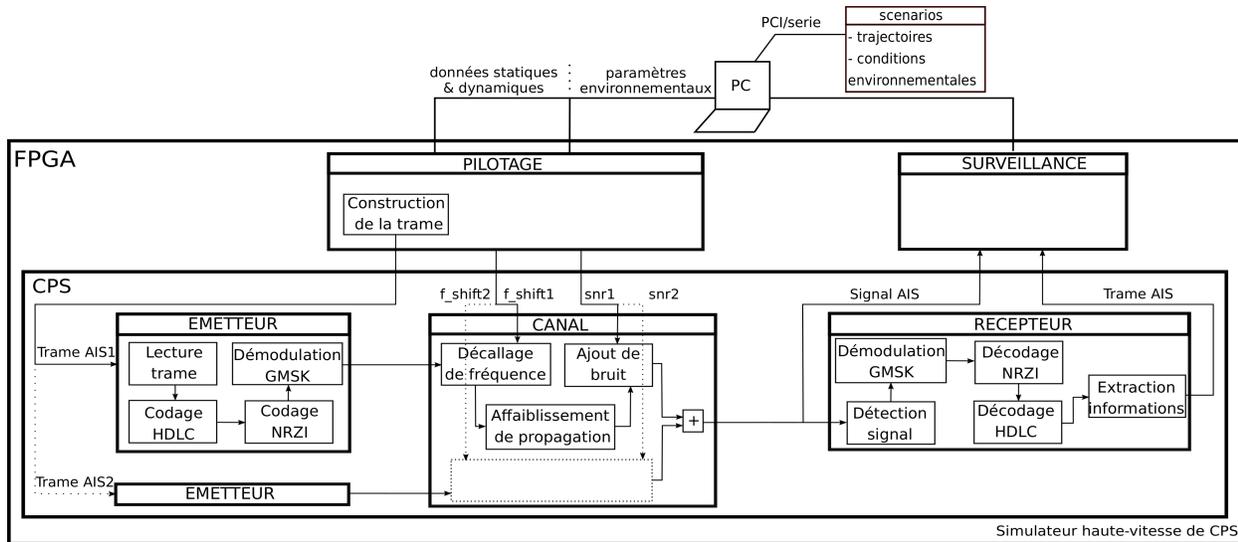


FIGURE 2 – Modèle détaillé du CPS simulé sur FPGA.

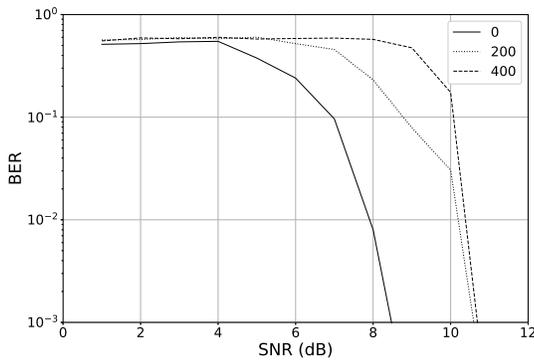


FIGURE 3 – Evolution du BER en fonction du SNR et du décalage de fréquence.

5 Discussion

Notre méthode d'émulation de CPS est limitée aux CPS dont l'architecture est un graphe orienté acyclique. En dehors de ce type d'architecture, Vitis HLS ne peut pas synthétiser le système. Par exemple, dans notre cas d'application, les transpondeurs AIS communiquent entre eux selon le protocole TDMA (Time-division multiple access) dont l'architecture est un graphe non orienté avec autant de noeuds que de bateaux. Ainsi, ce protocole ne peut pas être émulé par notre méthode. D'autres outils de synthèse de haut niveau, permettant une modélisation à base d'acteurs, sans limite dans la complexité de l'architecture modélisée, utilisant par exemple, une sémantique purement en flots de données, sont donc à expérimenter.

6 Conclusion

Dans cet article, nous avons proposé une méthode utilisant le FPGA comme plateforme d'émulation de systèmes cyber-

physiques complexes. Notre étude de cas porte sur l'émulation d'un ensemble de transpondeurs AIS fonctionnant dans un environnement maritime simulé. Les progrès des outils de HLS et les capacités croissantes des FPGAs rendent une telle méthode possible. Par rapport à une émulation purement logicielle, nous montrons des gains d'accélération remarquables ($\times 654$). Nous soulignons, cependant, certaines limites à notre méthode, qui nécessitent des approches alternatives pour pouvoir simuler une large gamme d'architectures. En somme, notre article démontre que le FPGA peut être considéré comme une plateforme de choix pour l'émulation de systèmes cyber-physiques.

Références

- [1] T. Grimm, B. Janssen, O. Navarro, and M. Hubner. *The value of fpgas as reconfigurable hardware enabling cyber-physical systems*. 2015 IEEE 20th Conf. on Emerging Technologies & Factory Automat. (ETFA), pp. 1-8.
- [2] S. Mojlish, N. Erdogan, D. Levine, and A. Davoudi. *Review of hardware platforms for real-time simulation of electric machines*. IEEE Trans. on Transp. Electrification, vol. 3, no. 1, pp. 130-146, 2017.
- [3] B. Miller, F. Vahid, and T. Givargis. *Meds : Mockup electronic data sheets for automated testing of cyber-physical systems using digital mockups*. 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1417-1420, IEEE
- [4] F. Katsilieris, P. Braca, and S. Coraluppi. *Detection of malicious ais position spoofing by exploiting radar information*. in Proc. of the 16th Int. Conf. on information fusion, pp. 1196-1203, IEEE, 2013.