

Mise en œuvre d'un algorithme de gestion de collision dans un réseau sur puce multi-FPGA

ATEF DORAI¹, VIRGINIE FRESSE¹, NABILA FILALI MERCHAOU²

¹ Laboratoire Hubert Curien UMR CNRS 5516

18 rue du Professeur Benoît Luras, 42000 Saint-Étienne, France

² Unité de Métrologie et des Systèmes Energétiques (UR11ES59)-ENIM

Rue Ibn el Jazzar, Skanès-5019 Monastir, Tunisie

¹virginie.fresse@univ-st-etienne.fr

Résumé - Les plateformes multi-FPGA sont les solutions les plus appropriées pour l'émulation de large SoC (System-On-Chip) basé sur des structures de communication de type réseaux sur puce (NoC : Network-On-Chip). Le déploiement d'un NoC sur plusieurs FPGAs nécessite la mise en place d'interfaces pour la communication inter-FPGA. Le nombre et le type de protocoles de communication disponibles sur les plateformes de prototypage limitent la bande passante du trafic inter-FPGA. Les collisions deviennent alors le problème majeur dans l'émulation de large SoC. Dans ce papier, nous proposons de mettre en œuvre un algorithme de gestion de collision inter-FPGA. L'algorithme proposé est extrait des réseaux informatiques (réseaux sans fils et Wifi) avec des adaptations pour correspondre aux besoins des NoCs sur FPGA. Des implantations de cet algorithme dans un NoC existant mettent en avant l'intérêt d'un tel algorithme avec un nombre de ressources supplémentaires étant faible.

Abstract - Multi-FPGA platforms are the most appropriate experimental solution to emulate large size of SoC (System on Chip) based on a NoC (Network on Chip). The deployment of the NoC onto several FPGAs requires the use of inter-FPGA communication links. The number and performance of external links restrict the bandwidth of the communication. Collision is considered as the major problem for NoC implemented in Multi-FPGA architecture. In this paper, we propose a solution for the collision detection due to the use of inter-FPGA links. The collision detection algorithm is extracted from wireless networking and wifi and then adapted to multi-FPGA and NoC structure. Implementations on FPGA show that the proposed algorithm provides an appropriate solution with a small number of extra resources.

1 Introduction

Les réseaux sur puce (NoC : Network On Chip) sont actuellement les structures de communications les plus adaptées pour les systèmes embarqués multicœurs. Le nombre de cœurs (ou d'IPs) devenant de plus en plus important, il est nécessaire de mettre en œuvre des tailles de NoC (s'exprimant en nombre de nœuds) de plus en plus grandes. Dans un contexte de prototypage, il n'est pas rare de déployer des structures de NoC sur plusieurs FPGAs, le nombre total de ressources du NoC et des cœurs de calcul étant nettement supérieur aux ressources d'un seul FPGA. La mise en œuvre nécessite alors d'utiliser les protocoles de communication disponibles sur les plateformes de prototypage. Certaines expérimentations ont permis de découper le NoC et de remplacer les liens physiques internes par des protocoles de communication externes [1][2][3]. Cette solution est possible dans le cas où la taille du NoC est petite et nécessite dans ce cas de remplacer quelques liens physiques par des communications externes. Le nombre de protocoles étant limité et le nombre de routeurs du NoC étant élevé, il devient nécessaire de partager un protocole de communication sur plusieurs routeurs [4]. Des goulots d'étranglements apparaissent, ralentissant fortement les performances du système déployé.

Dans ce travail, nous proposons de mettre en œuvre un algorithme de gestion de collision permettant de gérer les flux inter-FPGA. Cet algorithme est basé sur

ceux mis en œuvre dans les réseaux informatiques notamment les architectures infrastructures et ad-hoc. Des développements d'algorithme CSMA/CA (CA : Collision Avoidance) ont déjà été mis en œuvre sur FPGA [5][6] ayant pour objectif de simuler le réseau Wifi dans les applications de télécommunication. L'originalité de notre travail est la mise en œuvre d'un algorithme de gestion de collision, conçu à la base pour les réseaux informatiques, avec des modifications et des ajustements nécessaires pour l'adapter aux congestions des communications inter-FPGA liés aux NoCs. La proposition de la nouvelle solution dédiée aux systèmes et réseaux sur puce est ensuite développée et implantée sur FPGA.

Le papier comporte 5 sections. La section 2 décrit le principe des algorithmes de gestion de collision dans les réseaux informatiques. La section 3 présente le concept du déploiement des NoCs dans des plateformes multi-FPGA et le développement des blocs dédiés à la gestion de collision. La section 4 décrit les expérimentations réalisées, notamment les analyses de ressources des blocs IP et leurs intégration dans un NoC existant. Une conclusion et des perspectives achèveront ce papier dans la section 5.

2 Algorithme de gestion de collision dans les réseaux informatiques

Dans les réseaux informatiques, plusieurs types d'algorithmes de gestion de collision sont développés

pour s'adapter aux différentes topologies des réseaux informatiques. Sur les réseaux sans fils sont utilisés les topologies ad-hoc et les topologies infrastructures. Les topologies ad-hoc permettent aux machines de se connecter entre elles, ce qui constitue une communication point à point sans point d'accès (chaque machine jouant le rôle d'un client et d'un accès). Pour les topologies infrastructures, un réseau dorsal (backbone) est mis en œuvre autour duquel sont connectés les points d'accès (AP signifiant Acces Point) comme présenté en figure 1. Le protocole Wifi (IEEE802.11) intègre deux méthodes d'accès qui peuvent être utilisées conjointement, seules ou en alternance. La première méthode est une fonction de coordination DCF (Distributed Coordination Function) qui permet à toutes les machines d'accéder au médium lorsque celui-ci est libre. Cette méthode, similaire à la méthode de best effort permet des accès aléatoires et égalitaires mais non garantis. Elle peut être utilisée sur les topologies infrastructures et ad-hoc. La seconde méthode est une fonction de coordination du point d'accès PCF (Point Coordination Function), qui permet des accès à tour de rôle et n'est utilisée que sur la topologie infrastructure. Les accès sont ainsi assurés mais les performances sont plus faibles. Le travail proposé s'appuie sur la méthode d'accès DCF car elle correspond plus aux besoins des NoCs et permet aux routeurs d'accéder au point d'accès dès que des paquets sont présents.

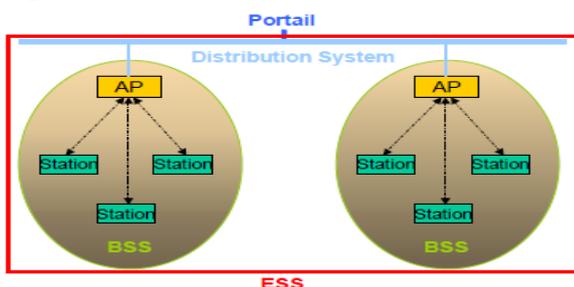


Figure 1 : Topologie infrastructure dans les réseaux informatiques

La fonction DCF utilise le protocole CSMA/CA pour accéder au canal de transmission. Dans ce mode, chaque station vérifie si le canal est libre avant l'envoi du paquet. S'il est libre pendant une période DIFS (Distributed Inter-Frame Space) alors le paquet est envoyé. Par contre, s'il est occupé, la transmission est retardée. Dans le cas où le canal reste occupé, l'algorithme de Backoff sur la station en attente est lancé afin de réduire les risques de contention. Cet algorithme définit un temps d'attente, appelé Backoff_Timer, calculé aléatoirement avant la réémission des paquets dans le nœud AP lorsque celui-ci sera disponible.

Les architectures dans les réseaux informatiques (stations vers canal) présentent une forte analogie avec les structures des NoCs. D'où l'utilité de mettre en œuvre des protocoles de communication inter-FPGA utilisés pour la communication entre plusieurs NoCs déployés sur plusieurs FPGAs. Dès lors, nous proposons de mettre en œuvre un algorithme de gestion de

collision sur FPGA à partir des algorithmes utilisés dans les réseaux informatiques pour permettre de gérer au mieux les paquets que les routeurs souhaitent envoyer.

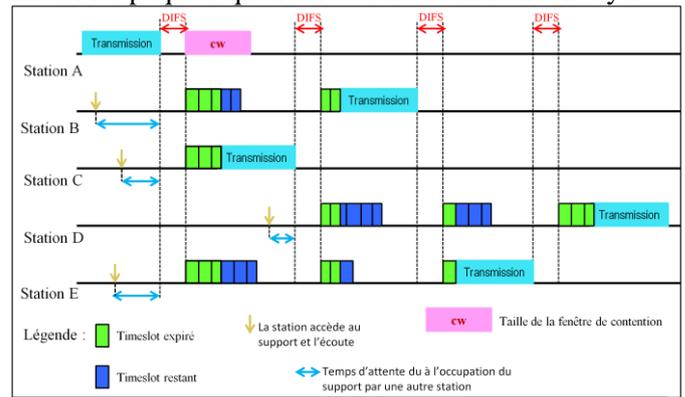


Figure 2 : Principe de l'algorithme Backoff pour un réseau à 5 stations

3 Mise en œuvre de l'algorithme de gestion de collision dans un réseau NoC

L'objectif est de gérer les goulots d'étranglement liés au nombre limité de protocoles de communication disponibles entre les plateformes FPGAs. Similairement aux réseaux informatiques, on considère donc que l'IP gérant la communication inter-FPGA constitue le point d'accès (AP).

3.1 Déploiement du NoC sur multi-FPGA

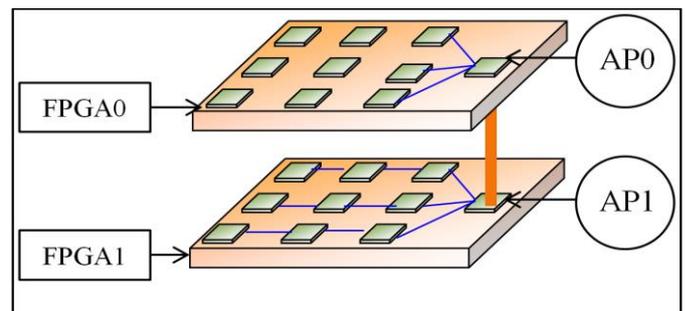


Figure 3 : Structure de communication sur plusieurs FPGAs. Plusieurs techniques de déploiement existant, nous considérons les postulats suivants :

- Chaque FPGA intègre un NoC synchrone qui possède sa propre fréquence.
- Il y a un AP, positionné par l'utilisateur, pour chaque FPGA.
- Un ou plusieurs routeurs du NoC peuvent être connectés à l'AP.
- Tous les protocoles disponibles sur une carte de prototypage peuvent être intégrés dans la plateforme finale, de type GALS (Globally Asynchrone Locally Synchrone).

Le déploiement du NoC permet d'obtenir une structure identique à celle illustrée par la figure 3.

3.2 Structure de communication inter-FPGA

L'intégration d'un protocole de communication inter-FPGA et de l'algorithme de gestion de collision nécessite l'ajout d'un certain nombre de blocs entre les routeurs du NoC et l'AP. Ces blocs sont décrits en VHDL synthétisable avec des paramètres génériques

pour s'adapter à différentes structures de NoC. Les IPs développés sont :

- Un point d'accès (AP) : Le point d'accès est chargé de vérifier que le protocole de communication est disponible et d'envoyer les paquets sur le protocole externe.
- Un bloc de contrôle : qui constitue le cœur de l'architecture infrastructure. Il permet de gérer respectivement les demandes d'accès multiples (émission et réception) via l'arbitre, de multiplexer et de démultiplexer les signaux de contrôle et de données entre les routeurs spécifiques et le point d'accès (AP). L'algorithme de routage utilisé est de type statique.
- Des routeurs spécifiques : sont des routeurs spécifiques situés à côté du NoC existant. Chaque routeur contient une unité de transmission tx, une unité de réception rx, une FIFO et des IP Backoff_Timer. La transmission des données entre les routeurs spécifiques et l'AP est de type parallèle/série.
- Un synchroniseur double bascule : La communication dans un système GALS nécessite l'ajout d'une interface de synchronisation utilisant deux bascules en série synchronisées, placées entre le NoC synchrone et les routeurs spécifiques dédiés à la communication inter-FPGA [11].

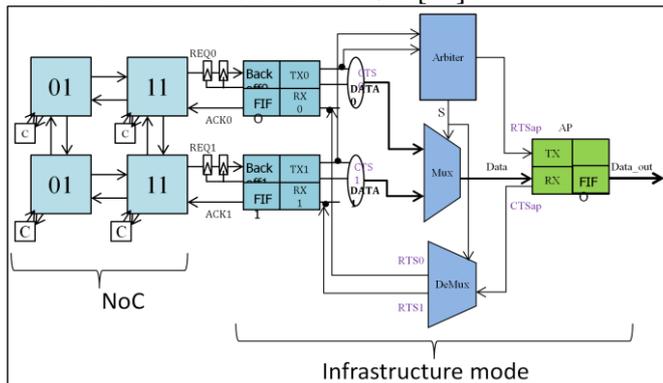


Figure 4 : Structure d'un NoC intégrant l'algorithme de gestion de collision adaptée au contexte NoC multi-FPGA

L'architecture globale étudiée, donné en figure 4, est composée de deux étages : le réseau NoC (à gauche) et l'architecture infrastructure (à droite). L'algorithme de gestion de collision développé concerne la partie « infrastructure architecture » sur laquelle des insertions d'IPs ont été faites. Ainsi, à partir du nombre de routeurs par ligne dans le réseau NoC existant, on fixe le nombre de routeurs spécifiques qui doivent être connectés via le bloc de contrôle à l'AP. L'utilisateur peut définir lui-même les constantes Backoff_Timer associées à chaque routeur dans un package de configuration. Dans d'autres cas, il est possible d'insérer un IP dédié à la génération aléatoire de ces temporisateurs pour chaque routeur.

3.3 Blocs du routeur spécifique

La transmission des données entre les routeurs spécifiques et l'AP est de type parallèle/série. Les blocs dédiés à la gestion de collision sont les suivants :

- IPs pour la gestion de l'algorithme de Backoff : l'algorithme a été adapté aux contraintes d'implantation et aux structures de NoC. Il permet de définir un temps d'attente pour l'accès à l'AP lorsque le protocole est occupé. La constante **Backoff_timer** utilisée pour attendre l'accès peut être une constante calculée de manière aléatoire ou définie par l'utilisateur à la suite d'une analyse de trafic sur le protocole inter-FPGA externe. La variation de cette constante est extraite de ce qui est utilisé dans les réseaux informatiques. Le principe est donné en figure 5.
- Transmit Unit (tx) : chargé pour transmettre les paquets vers l'AP (figure 6).
- Receive Unit (rx) : chargé pour recevoir les paquets venant de l'AP (figure 7).

```

Step 0: set Backoff_Timer to the initial value
        Backoff_Timer = CW_min * TimeSlot
Step 1: while Backoff_Timer ≠ 0 do
        For each timeSlot
            if channel is idle then
                Backoff_Timer = Backoff_Timer - 1
                if Backoff_Timer = 0 then
                    Send and CW = CW_min
                else
                    Backoff_Timer = Backoff_Timer - 1
            else
                Backoff_Timer = 2 * CW_min * TimeSlot - 1
        if CW = CW_max then
            Go to Step 1
        Stop
    
```

Figure 5 : Principe de l'algorithme Backoff adapté aux NoCs

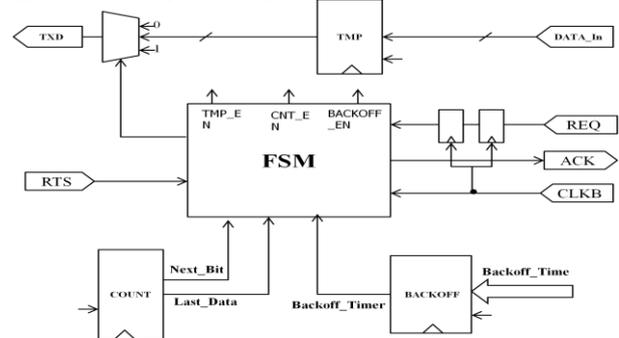


Figure 6 : Unité de transmission « tx »

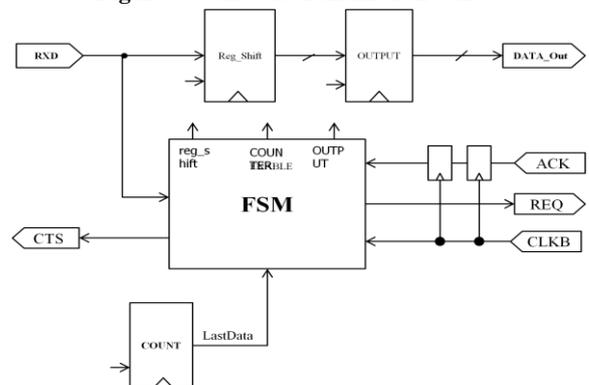


Figure 7 : Unité de réception « rx »

4 Expérimentation

L'architecture infrastructure, intégrée au NoC Hermes [10], est implémentée sur une plateforme

ML506 basée sur un FPGA Virtex 5 de Xilinx [12]. Les premières expérimentations portent sur l'architecture infrastructure seule avec un nombre de routeurs variant de 1 à 16. Les ressources en nombre de registres sont données en figure 8.

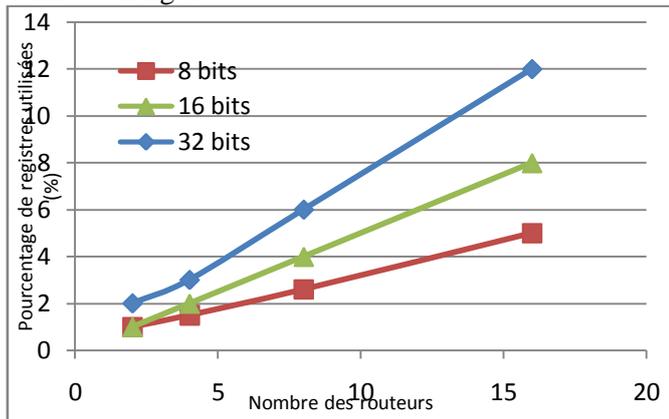


Figure 8 : Pourcentage de registres utilisés sur Xilinx Virtex 5 pour différents tailles de l'architecture infrastructure

Le pourcentage de ressources en registres nécessaires est faible comparé aux ressources disponibles actuellement sur les FPGAs. Elles représentent respectivement moins de 6%, 8% et 12% pour un NoC ayant des flits de tailles 8, 16 et 32 bits. Ces estimations sont proches pour les ressources de type LUTs. La seconde expérimentation porte sur le NoC Hermes (avec trois exemples de tailles : 2x2, 4x2 et 4x4) avec et sans l'architecture infrastructure. Le taux d'occupation en registre est donné dans la figure 9 (le nombre en pourcent de LUTs suit la même évolution que le nombre de registres).

On observe que le nombre de ressources nécessaires à l'architecture infrastructure augmente de manière significative lorsque la taille du NoC augmente. Le système composé d'un NoC 4x4 avec l'architecture infrastructure consomme 50% de plus en termes de ressources qu'un système basé sur un NoC 4x2.

Le nombre de registres et le nombre de LUTs restent très proches (Figure 10).

Ainsi pour des tailles de NoC importantes, nécessitant le déploiement sur plusieurs FPGAs. Les ressources matérielles supplémentaires (de l'architecture infrastructure) restent faibles comparées aux ressources du NoC.

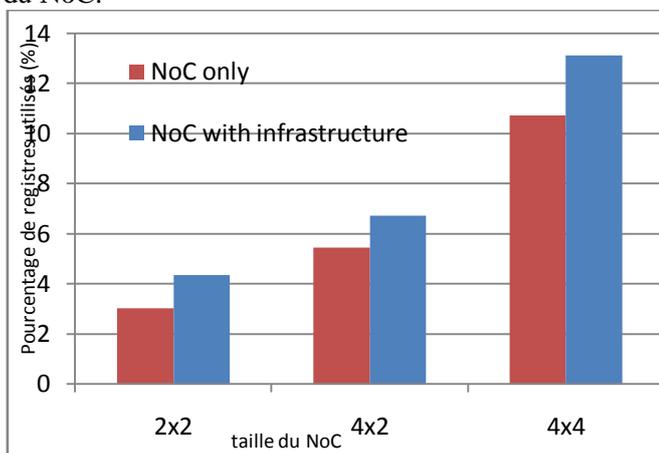


Figure 9 : pourcentage de registres utilisés par chaque architecture de NoC

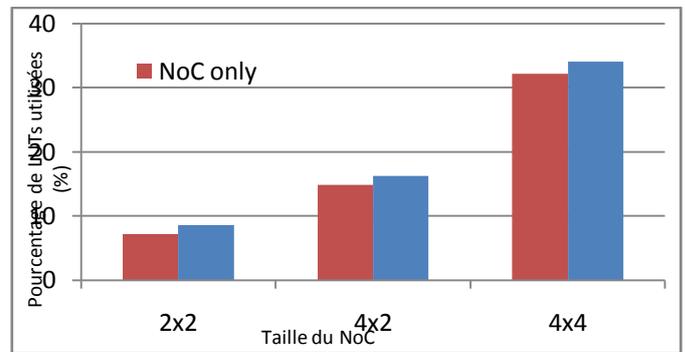


Figure 10 : pourcentage de LUTs utilisés par chaque architecture de NoC

5 Conclusion

Ce papier présente le principe et la mise en œuvre d'un algorithme de gestion de collision basé sur l'algorithme de Backoff pour le déploiement d'un NoC sur une plateforme multi-FPGA. Les blocs IPs nécessaires ont été inspirés des réseaux informatiques avec des adaptations pour répondre aux contraintes des NoCs. Les résultats des expérimentations mettent en évidence que l'utilisation de cet algorithme permet une gestion optimisée des protocoles de communication inter-FPGA. Les ressources matérielles exploitées pour l'architecture infrastructure restent très faibles. Une amélioration de l'algorithme de gestion de collision est une des perspectives envisagées.

6 Remerciements

Ce travail est supporté par la Région Rhône-Alpes.

7 Références

- [1] K. M. Abdellah-Medjadji, B. Senouci, F. Petrot, "Large Scale On-Chip Networks : An Accurate Multi-FPGA Emulation Platform, " 11th EuroMicro Conference on Digital System Design Architectures, Methods and Tools, pp. 3-9, 2008.
- [2] L. Xinyu, O. Hammami "Multi-FPGA emulation of a 48-cores multiprocessor with NOC" 3rd international Design and test workshop, pp 205, December, 2008, Tunisia.
- [3] Virginie Fresse, Junyan Tan, Frederic Rousseau. From Mono-FPGA to Multi-FPGA Emulation Platform for NoC Performance Evaluations.
- [4] C. Seiculescu, S.Murali, L. Benini: "SunFloor 3D: A Tool for Networks on Chip Topology Synthesis for 3-D Systems on Chips", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.29, December 2010.
- [5] B.S. Feero, P.P. Pande: "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation", IEEE Transactions on Computers, Vol. 58, January, 2009.
- [6] W.L. Pang, K. W. Chew, Florence Choong, E.S. Teoh. VHDL Modeling of the CSMA/CA.
- [7] S. Wang, H. Zhao, S. Huang, A. Hafid "Modeling FPGA-based IEEE 802.11 DCF", 7th International on mobile Ad-hoc and sensor Networks, Beijing, Chine, décembre 2011.
- [8] A.M. Bjavilkatti, "VHDL Modeling of Wi-Fi MAC Layer for Transmitter". IEEE International Advance Computing Conference, mars, 2009.
- [9] IEEE std 8802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. 1999.
- [10] Fernando Gehm Moraes, Ney Laert Vilar Calazans, Aline Vieira de Mello, Leandro Heleno Möller, Luciano Copello Ost. HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip.
- [11] Y. Semiat et R. Ginosar, "Timing Measurements of Synchronization Circuits", Proceedings of the Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC'03, Vancouver, Canada.
- [12] <http://www.xilinx.com/>

