

# Implémentation par Assemblage d'IP d'Architectures Massivement Parallèles

Mouna Baklouti<sup>1,2,3</sup>, Sébastien Le Beux<sup>2</sup>, Philippe Marquet<sup>2</sup>, Jean-Luc Dekeyser<sup>2</sup>, and Mohamed Abid<sup>1</sup>

<sup>1</sup> Laboratoire CES, École Nationale d'Ingénieurs de Sfax, Tunisie

<sup>2</sup> LIFL, INRIA Lille-Nord Europe, Université de Lille, France

<sup>3</sup> [mouna.baklouti@lifl.fr](mailto:mouna.baklouti@lifl.fr)

**Abstract.** De nos jours, l'évolution de la technologie des semi-conducteurs permet l'intégration de milliards de transistors sur une seule puce. Cela nous permet alors de concevoir des systèmes de plus en plus complexes et variés. De plus, avec la complexité et l'hétérogénéité grandissantes des systèmes sur silicium, des composants préexistants, appelés IP, sont de plus en plus intégrés et réutilisés pour la conception de tels systèmes. Dans ce papier, nous montrons la faisabilité de la conception d'une architecture massivement parallèle nommée MppSoC sur FPGA et présentons notre proposition d'une méthodologie d'assemblage d'IP pour une telle conception.

## 1 Contexte : les architectures SIMD sur puce

Nous ciblons dans nos travaux les applications de traitement de signal intensif et plus spécifiquement le calcul data-parallèle. Les systèmes embarqués massivement parallèles, particulièrement à architecture SIMD (Single Instruction Multiple Data), sont devenus une solution incontournable pour l'exécution de telles applications. Typiquement, une machine SIMD est une grille de processeurs appelés processeurs élémentaires (PE) connectés par un réseau de topologie simple. Chaque processeur exécute la même instruction à chaque cycle d'une manière synchrone, orchestré par un processeur de contrôle unique appelé ACU (Array Control Unit). L'intérêt principal de nos travaux est de reconsidérer la faisabilité et les performances des machines massivement parallèles dans le cadre de leur intégration sur des architectures reconfigurables telles FPGA. L'intérêt suscité par les FPGA est essentiellement motivé par leur grande flexibilité (grâce à leur reprogrammabilité), la facilité qu'ils présentent dans la mise en œuvre des applications, etc. Pour faciliter et accélérer la conception, nous voulons suivre les nouvelles méthodologies d'intégration à savoir l'utilisation et la réutilisation, ou l'assemblage, de composants dits IP (Intellectual Property).

Dans ce papier, nous définissons une architecture SIMD massivement parallèle, nommée MppSoC (Massively parallel processor System on Chip). Un aspect de conception distingue cette proposition de celle des autres du fait de l'utilisation d'IP dans la conception offrant ainsi un gain appréciable en terme

de temps de développement. De même la réutilisation d'IP pour concevoir l'ACU et les PE est une nouvelle proposition dans le domaine SIMD. Le reste du papier est organisé comme suit : la section 2 présentera brièvement les travaux existants pour la conception des architectures massivement parallèles sur puce, et soulignera l'intérêt de la réutilisation d'IP et le besoin de standardiser la description de leurs interfaces facilitant ainsi l'assemblage. La section 3 sera dédiée la description de l'implémentation actuelle de MppSoC sur FPGA. La section 4 discutera des premiers résultats auxquels nous avons aboutis et de leurs possibles améliorations. Finalement, la section 5 expliquera les contributions attendues à plus long terme tout en exposant les perspectives de ces travaux.

## 2 État de l'art

Les machines SIMD sont apparues dès les années 60 avec le projet ILLIAC IV [1, 2], puis avec les Connection Machine [3] et Maspar [4], etc. De nos jours, ces machines répondent aux besoins de plusieurs applications émergentes telles que les applications de traitement de signal intensif. Par ailleurs, les avancées technologiques concernant l'intégration rendent le placement d'une grille SIMD complète sur une unique puce envisageable, facilitant la diffusion synchrone d'instructions à des fréquences élevées. Cette considérable évolution montre qu'il devient aujourd'hui économiquement intéressant de concevoir des machines SIMD avec des centaines de processeurs élémentaires et mémoires embarquées sur puce. De ce fait, des systèmes SIMD sur puce commencent à apparaître dans les dernières années.

L'équipe de H.-S. S Lee à Georgia Tech propose une architecture multi-processeur massivement parallèle, appelée PoD (Parallel on Die) se basant sur le modèle SIMD en vue d'une implémentation sur puce [5]. Cette architecture se compose d'un processeur hôte, d'un réseau SIMD de PE (8x8) et d'un réseau d'interconnexion PoD pour la communication inter-PE. Les PE sont des processeurs VLIW (Very Long Instruction Word) exécutant des instructions entières, SIMD et mémoire. La communication voisin à voisin est totalement contrôlée par le logiciel et ne nécessite ni arbitrage, ni routage. Ce travail ne traite pas le sujet de communication, du côté du matériel, qui est assez important dans les architectures parallèles et gère toute sa complexité en logiciel.

Le projet Co-MAP [6, 7], acronyme de Co-Design of Massively Parallel Embedded Processor Architectures, propose une plate-forme générique de processeurs reconfigurables dynamiquement. Le système comporte un ensemble de PE avec des mémoires locales de petites tailles et un réseau d'interconnexion régulier qui peut être reconfiguré dynamiquement. Les processeurs sont qualifiés de WPPE (Weakly Programmable Processing Element) à architecture VLIW et avec un jeu d'instructions réduit et spécifique aux applications de traitement de signal. Une première implémentation contenant 4x4 processeurs a été testée sur FPGA en utilisant une très simple configuration. Ce travail s'est limité à détailler la conception des réseaux d'interconnexions dynamiques pour les archi-

tectures massivement parallèles et il n'a pas traité la générnicité de tout le modèle architectural. De plus, l'architecture décrite ne suit pas le modèle SIMD.

La plupart des travaux existants n'exploitent pas les nouvelles méthodologies d'intégration à savoir la réutilisation d'IP. Ces IP devront être interopérables, flexibles et présentant des interfaces standards facilitant leur intégration. Nous nous intéressons alors à l'étude des standards et normes d'interfaces d'IPs actuels. Dans ce contexte, trois principaux organisations proposent des protocoles configurables, point à point, non spécifiques à un bus tel que : VSIA (Virtual Socket Interface Alliance), OCP-IP (Open Core Protocol-International Partnership) et SPIRIT (Structure for Packaging, Integrating, and Re-using IP within Tool flows).

OCP-IP est un consortium qui définit le protocole OCP. OCP [8] ressemble au protocole VCI (Virtual Component Interface), ajoutant de plus des signaux de contrôle et de test pour unifier toutes les communications entre les IP. OCP définit un protocole extensible, configurable et point à point utilisant les adaptateurs et les notions d'initiateurs et de cibles.

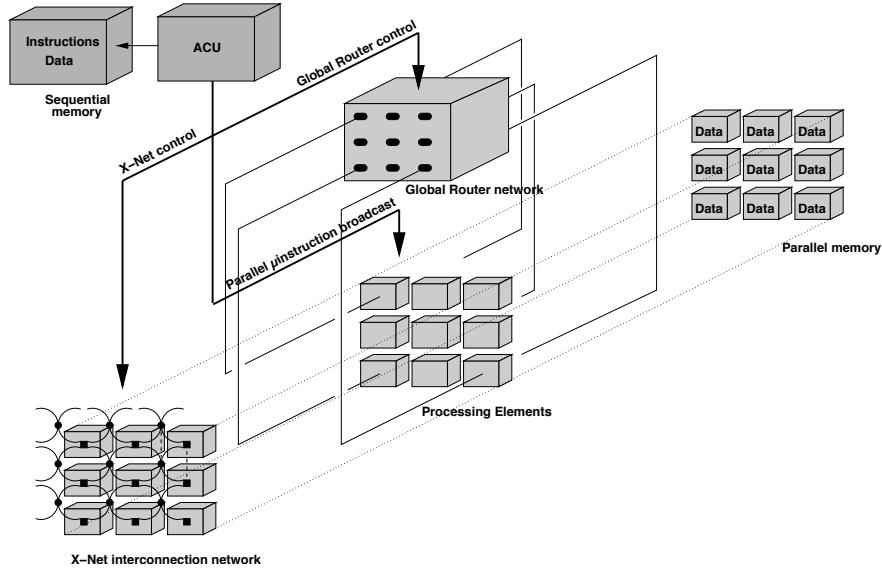
Le consortium SPIRIT [9] définit le standard IP-XACT. Il se base sur le format XML pour permettre de décrire des composants au niveau RTL et au niveau TLM. SPIRIT est en effet un mécanisme de spécification commun pour la description de la propriété intellectuelle, qui permet une configuration et une intégration automatisée à travers des outils plug-in. Jusqu'à maintenant, SPIRIT ne présente que la manière de modélisation d'un composant et ne décrit pas clairement son interface.

Notre objectif est donc d'adopter un standard pour la description des interfaces des IPs de notre architecture.

### 3 Première implémentation de MppSoC sur FPGA

MppSoC, inspiré de la MasPar, est un système multiprocesseur SIMD composé d'une grille 2-D de processeurs et mémoires connectés par un réseau de voisinage XNet et un routeur global mpNoC [10, 11] (Fig 1). L'ACU et les PE sont construits à partir du même IP processeur (miniMIPS [12]). Ce choix était principalement basé sur des raisons pragmatiques : le MIPS est un processeur simple, standard et bien établi. Quelques mineures additions sont ajoutées à ce processeur pour concevoir l'ACU, alors que sa partie de décodage est supprimée dans le PE assurant ainsi une meilleure intégration sur puce et réduisant la consommation. Donc en utilisant le même IP, le haut coût de conception lié aux machines SIMD est considérablement réduit. Cette solution reste complètement une nouvelle approche dans le domaine de conception SIMD.

Pour une première implémentation partielle sur FPGA, nous avons visé une architecture MppSoC simple, considérée comme une étude de faisabilité de l'architecture conçue. Le prototype actuellement développé (Fig 2) intègre 16 PE s'exécutant parallèlement. Il n'y a pas de réseau de communication intégré. Le nombre de PE est paramétrable en deux dimensions pour former une grille. Il est possible aussi de paramétriser les dimensions afin d'obtenir une ligne de PE.



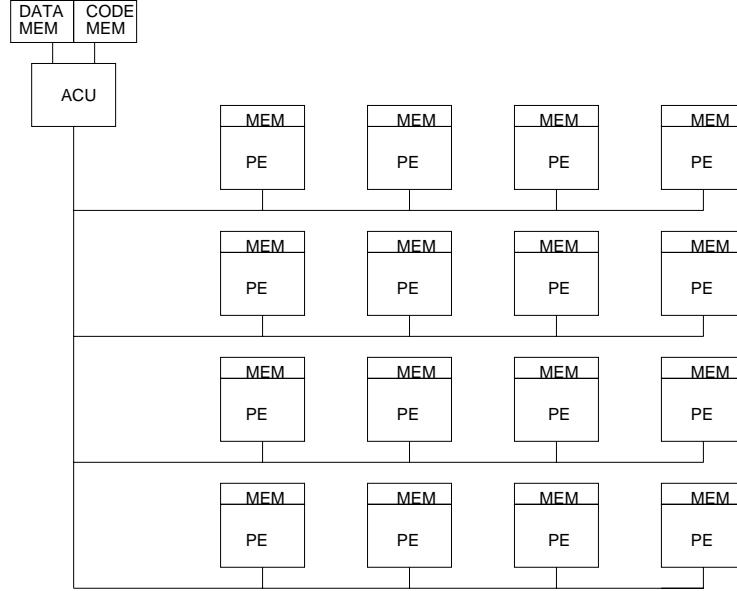
**Fig. 1.** Architecture MppSoC [11]

Les connexions entre l'ACU et les PE sont des fils qui relient l'étage de décodage de l'ACU aux étages d'exécution des PE. La seule autre connexion est assurée par les mémoires. L'ACU peut accéder à toutes les mémoires des PE directement par une plage d'adresses distinctes.

Ce prototype est hiérarchique comme le montre la Fig 3. En effet, les éléments de la grille sont inclus dans une unité hiérarchique appelé Unité Élémentaire, contenant chacune un PE et ce qui lui est local (mémoire + sélecteurs). Un autre niveau de hiérarchie, appelé Bloc d'Unités Élémentaires, correspond à une ligne dans la grille d'unités élémentaires. Ils sont créés afin de limiter le synthétiseur dans ses optimisations (lors de la fusion de registres par exemple). Ces optimisations indésirables créent de la logique centrale et augmentent la difficulté de routage. Les composants implémentés sur FPGA seront présentés dans les paragraphes suivants.

### 3.1 Implémentation de l'ACU

L'ACU contient un pipeline de 5 étages. Seul l'étage de décodage des instructions a été modifié permettant le décodage des instructions vectorielles et le support des instructions multi-cycle. L'implémentation est générique et peut être réutilisée pour d'autres instructions en les ajoutant dans la table d'instructions. L'interface entre cet étage et l'étage d'exécution de l'ACU ou des PE n'est pas la même. L'interface vers les PE contient moins de fils. Ce choix a pour but de diminuer le nombre de connexions entre l'ACU et les PE afin de diminuer la complexité du routage et augmenter la fréquence du système. L'ACU possède 4



**Fig. 2.** Prototype FPGA de MppSoC

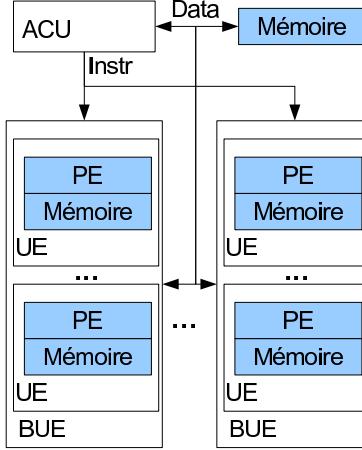
ko de mémoire, comme chaque PE, implémentée dans les blocs mémoires M4k du FPGA. L'ACU peut accéder directement à la mémoire de tous les PE par un bus. Ce dernier facilite le transfert de données entre l'ACU et les PE ; en contre partie il est coûteux en terme de nombre de fils.

### 3.2 Implémentation du PE

Les PE sont réduits par rapport à l'ACU. Le pipeline des PE ne contient en fait que les trois derniers étages. Une particularité intéressante est que les modules du pipeline sont les mêmes tant pour l'ACU que pour les PE. Cependant, plusieurs signaux d'entrées non utilisés par les PE sont remis à zéro afin de les retirer. Les mémoires des PE sont accessibles par les PE et par l'ACU. Elles contiennent donc des multiplexeurs à l'entrée de leurs ports, permettant de choisir entre des requêtes ACU ou PE. Les PE se caractérisent par un bit d'activité qui contrôle le dernier étage du pipeline (write-back). En fait, si un PE n'est pas actif, il annule l'écriture dans les registres d'instruction et annule les opérations mémoires.

## 4 Analyse des premiers travaux

Le prototype a été implémenté sur un FPGA Altera Stratix2 de 60k éléments logiques. L'implémentation actuelle est un code VHDL qui peut être utilisé par les outils Quartus II et ModelSim Altera [13]. Cette implémentation est une



**Fig. 3.** Hiérarchie de MppSoC

expérience pratique pour la réutilisation d'IP, nouvelle approche dans le domaine SIMD.

Le prototype implémenté génère une configuration FPGA paramétrée par le nombre des PE et la taille mémoire reliée à chaque PE : une application peut nécessiter un grand nombre de PE avec une petite taille mémoire ou bien un petit nombre de PE avec une grande taille mémoire. Une fréquence de fonctionnement 50 Mhz a été atteinte pour cette première configuration de MppSoC. Par ailleurs, une analyse de l'utilisation de la surface FPGA montre qu'un PE est beaucoup plus petit que l'ACU, que l'étage de décodage est le plus grand étage de l'ACU, et que l'étage d'exécution (950 LEs pour l'ACU, 1090 pour le PE) consomme plus de surface sur le FPGA. La problématique de connexion est aussi adressée dans MppSoC. En effet, en analysant les signaux de l'architecture nous avons remarqué que ceux permettant à l'ACU d'accéder aux mémoires des PE représentent 63 % de la totalité des signaux ce qui est très élevé. Par conséquent, cette connexion sera reconstruite dans les futures conceptions, spécialement avec l'intégration d'une future implémentation de mpNoC permettant d'assurer le transfert de données entre l'ACU et les mémoires des PE. Il est à noter que ces calculs sont valides pour des FPGA sans logique à trois états. Si on vise un ASIC ou un futur FPGA qui supporterait la logique à trois états, on peut réduire les signaux en diminuant la taille du bus de données entre l'ACU et les PE tout en utilisant des signaux bidirectionnels. Ce prototype nécessite de même plusieurs autres optimisations et améliorations à faire en terme de performance et surface. Il serait intéressant par exemple de séparer le bus mémoire de l'ACU vers sa mémoire du bus mémoire de l'ACU vers les mémoires des PE afin de doubler l'espace d'adressage et simplifier la synthèse. Le FPGA utilisé actuellement permet d'intégrer 16 PE ou même 32 PE tout en diminuant leurs tailles ce qui n'est pas intéressant vu que nous voulons intégrer aussi les deux réseaux de communications. Il faut penser alors à choisir d'autres types de FPGA plus

appropriés à l'architecture MppSoC. De plus, le processeur utilisé ne supporte ni la division ni les opérations à virgules flottantes qu'il faut ajouter dans les futures implémentations. A part toutes ces limitations, il en existe d'autres à résoudre dans le prototype actuel.

## 5 Directions de recherches

La conception et l'intégration des machines massivement parallèles sur une puce, tel MppSoC, sont considérées comme un levier permettant de répondre aux besoins des systèmes embarqués pour le calcul à hautes performances. Notre principal objectif est de réduire le coût et le temps de conception de ces systèmes afin d'obtenir des circuits fiables et performants. Tout ceci doit être construit selon une méthodologie de conception favorisant l'abstraction de l'architecture, la validation et l'utilisation des standards et facilitant la génération d'une description de bas niveau d'une architecture.

De ce fait, nous allons utiliser les méthodologies actuelles de conception des systèmes qui promeuvent les approches à base de composants et exploitent l'utilisation d'IP et d'interfaces standardisées. Ce contexte doit permettre d'alléger les coûts de conception d'un processeur dédié en vue de son intégration dans MppSoC. Le premier axe de nos travaux s'intéressera à extraire, normaliser et concevoir les IPs. Nous distinguons différents types d'IPs pour MppSoC typiquement des IP processeurs, des IP contrôleur, des IP mémoire et des IP réseaux de communication. Ces différents IPs peuvent se baser sur les IPs existantes sur le marché après modifications tel est l'exemple d'un IP processeur qui nécessite de supporter des instructions séquentielles et parallèles pour pouvoir être utilisé dans MppSoC. Ainsi, une autre implémentation a été faite en restructurant l'architecture actuelle de MppSoC. Nous avons commencé par découper alors l'architecture en des IPs notamment les deux types de processeurs ACU et UE puis reconstruire l'implémentation par assemblage de ces IPs. La conception à base de réutilisation d'IPs est incontournable pour répondre aux exigences de diverses applications. Cette expérimentation nous a permis d'accélérer la conception mais aussi de s'apercevoir de la difficulté à gérer l'interconnexion entre les différents IP.

L'implémentation actuelle n'inclut aucun réseau de communication entre les composants. Elle sera alors reconsidérée dans les futures conceptions par l'implémentation et l'intégration des IP Xnet et mpNoC. L'IP mpNoC, par exemple, sera conue telle que la topologie de son réseau d'interconnexion sera reconfigurable dépendant de plusieurs paramètres tels que la performance, la possibilité d'intégration et le nombre de PE. De plus, l'ajout de ces IPs nécessite une extension du jeu d'instructions pour supporter les instructions de Xnet et de mpNoC. Nous sommes confrontés alors à différents choix d'implémentation logiciels/matériels rigides parfois de point de vue matériel mais qui peuvent se compenser par une flexibilité logicielle. Il est donc nécessaire d'adopter une solution appropriée aux contraintes de performances. Cette conception, par assemblage d'IP, serait efficace et intéressante si tous les composants de l'architecture puis-

sent communiquer et s’interfacer facilement entre eux. En vue d’un assemblage de composants pour une configuration donnée de MppSoC il s’agit de définir des interfaces et de pouvoir réaliser les interconnexions indépendamment de la configuration du système. Nous procérons alors à une normalisation des interfaces des IP. Une étude est déjà en cours consistant à voir comment et quel standard utiliser pour décrire d’une manière générique les interfaces des IPs.

Nos travaux s’orienteront par suite à concevoir une machine ad-hoc en fonction de l’application. Nous voulons en fait définir une architecture générique afin de déduire une solution taillée selon les besoins d’un ensemble d’applications en terme de ressources de calcul, de mémorisation et de communication. Le deuxième axe serait donc de définir une manière d’assemblage intelligent des IP afin d’avoir une architecture générique. Nous pourrons construire une bibliothèque d’IP dédiée pour notre architecture parallèle et assembler après une combinaison choisie de blocs selon les besoins pour développer rapidement MppSoC et réduire les délais de conception. Il faut en effet une nouvelle méthodologie d’interconnexion pour développer des systèmes sur puce contenant un milliard de transistors ou plus, organisés en centaines, voire en milliers de blocs IP.

Nous notons que l’implantation FPGA actuelle a été réalisée manuellement : les choses sont à reprendre si le modèle de l’architecture varie ou si la configuration change en fonction des besoins des applications envisagées sur la machine, ce qui présente une tâche lourde lors de la conception. Une configuration donnée d’un système MppSoC décrira les composants devant être utilisés et les paramètres de dimensionnement de la machine (nombre de processeurs, taille mémoire associée à chaque processeur, éventuellement largeur du/des réseaux, etc.). L’utilisateur aura le choix d’intégrer les IPs prédéfinis et nécessaires pour son architecture, piloté par les exigences de son application. Nous voulons développer, pour cette fin, des outils capables de générer une bonne configuration de la machine taillée aux besoins de l’application. En effet, on trouve des applications qui se basent principalement dans leur exécution sur des communications régulières, il serait alors intéressant d’utiliser dans l’architecture le réseau Xnet plutôt que mpNoC. Autrement, si l’application nécessite des communications irrégulières il faut penser à l’utilisation de mpNoC. De même d’autres applications requièrent peu de calculs mais par contre de grandes quantités de données, ce qui nous amène à penser à utiliser un relativement petit nombre de PE avec une importante quantité de mémoire, etc.

## 6 Conclusion

Dans ce papier, nous avons défini un nouveau concept de système SIMD appelé MppSoC. Cette conception se base sur l’utilisation d’IP, nouvelle proposition dans le domaine SIMD. Une première implémentation, partielle, sur FPGA a été aussi validée. Elle est paramétrable en nombre de PE et la taille mémoire liée à chaque PE selon les exigences des applications. Les travaux futurs consisteront à implémenter et intégrer les réseaux de communication dans l’architecture actuelle. Ils se focaliseront aussi sur la manière de pouvoir concevoir une im-

plantation sur FPGA générique comportant différents composants, sur la base de l'utilisation d'IP, d'une architecture massivement parallèle taillée aux besoins de l'application envisagée. Notre objectif est de proposer et mettre en œuvre une méthodologie ad hoc pour la production d'implantations FPGA de l'architecture MppSoC.

## References

1. Hord, R.M.: The Illiac IV the first supercomputer. Computer Science Press (1982)
2. Davis, R.L.: The Illiac IV processing element. IEEE Transactions on Computers **18** (1969) 800–816
3. Hillis, W.D.: The Connection Machine. The MIT Press, Cambridge, MA (1985) Traduction française, Masson, Paris, 1988.
4. Blank, T.: The MasPar MP-1 architecture. In: Proceedings of the IEEE Compcon Spring 1990, San Francisco, CA, IEEE Society Press (1990) 20–24
5. Woo, D., Fryman, J., Knies, A., Marsha Eng, Lee, H.: POD: A parallel-on-die architecture. In: 1th Annual Workshop on High Performance Embedded Computing, Lexington, MA (2007)
6. Kissler, D., Hannig, F., Kupriyanov, A., Teich, J.: A highly parameterizable parallel processor array architecture. In: IEEE Int'l Conf. on Field Programmable Technology, FPT 2006, Bangkok, Thailand (2006) 105–112
7. Kissler, D., Hannig, F., Kupriyanov, A., Teich, J.: Hardware cost analysis for weakly programmable processor arrays. In: Int'l Symposium on System-on-Chip, Tampere, Finland (2006) 105–112
8. OCP, I.P.: Open core protocol specification 2.0. <http://www.ocpip.org/> (2003)
9. Consortium, T.S.: Enabling innovative IP re-use and design automation (2007) <http://www.spiritconsortium.org/>.
10. Duquennoy, S., Le Beux, S., Marquet, P., Meftali, S., Dekeyser, J.L.: MpNoC design: Modeling and simulation. In: 15th IP Based SoC Design Conference (IP-SoC 2006), Grenoble, France (2006)
11. Marquet, P., Duquennoy, S., Le Beux, S., Meftali, S., Dekeyser, J.L.: Massively parallel processing on a chip. In: ACM Int'l Conf. on Computing Frontiers, Ischia, Italy (2007)
12. miniMIPS OpenCores: miniMIPS overview. <http://www.opencores.org/projects.cgi/web/minimips/> (2006)
13. Altera: Home page (2007) <http://www.altera.com/>.