

Résumé

La feuille de route d'ITRS Semi-conducteur prévoit que des centaines de processeurs seront nécessaires pour les futures générations du multiprocesseur (MPSoC). La modélisation des multiprocesseurs, le niveau adéquat d'abstraction (TLM, RTL), l'évaluation de la performance et l'exploration d'espace de conception, la vérification et la simulation ou l'émulation sont les sujets actuels de recherche. L'efficacité de conception qui est l'un des défis les plus importants, est un problème de recherche relativement nouveau et ouvert. Nous proposons d'améliorer l'efficacité de conception en augmentant la taille d'IP SSM, et en combinant les techniques d'extension rapide au niveau du système avec multi-FPGA émulateur.

Dans la thèse, avoir analysé et comparé les différentes méthodes pour la conception de NoC et de MPSoC, nous proposons une procédure automatique et multi-objective pour NoC au niveau TLM (Transaction Level Modeling). Les critères du timing et de surface du niveau RTL sont explorés mais non limités avec des TLM modèles du NoC dans NoCexplorer. Une méthodologie de la programmation linéaire est fournie comme solution au problème de l'organisation et du dimensionnement de eFPGA reconfigurable pour maximiser l'efficacité du NoC.

Notre contribution principale est la procédure automatique pour la conception de MPSoC à grande taille basée sur la réutilisation de SSM IP. Basée sur ce principe, une procédure de conception automatique pour des données parallèles et des traitements en pipeline est proposée pour l'application au traitement du signal sur le multiprocesseur avec NoC, utilisant l'application cryptographique au TDES (Triple Data Encryption Standard) comme un exemple. La synthèse de haut niveau est ajoutée à cette procédure pour la génération de hardware accélérateur, qui permet d'étudier le compromis entre la performance et la surface. OCP-IP NoC benchmarks sont exécutés sur notre multiprocesseur de 48 coeurs et de 672 coeurs pour l'évaluation de performance.

Tous les travaux réalisés dans cette thèse rendent possible MPSOC explorer, un projet industriel pour l'exploration de MPSoC à grand taille, soutenu par l'Union Européenne et le gouvernement français.

1. Introduction

La feuille de route d'ITRS Semi-conducteur prévoit que des centaines de processeurs seront nécessaires pour les futures générations du multiprocesseur (MPSoC). La densité croissante du dispositif permet exponentiellement plus de cœurs sur une seule puce. Les fabricants de processeur ont décalé vers la production des processeurs multi-cœurs pour respecter les contraintes de la puissance et du rafraîchissement tout en maintenant les avancées de performance d'exécution attendues avec chaque nouvelle génération de processeur. Le processeur à 8 cœurs de l'Intel est attendu d'ici 2009 et la performance de Tflops avec 80 cœurs en 45 nm technologie a déjà été démontrée. Des processeurs graphiques ont déjà des centaines de cœurs, tels que le récent GeForce 295 du NVIDIA avec 480 cœurs en 55 nm technologie. D'ailleurs, ITRS prévoit que la même tendance continuera également, et qu'une amélioration jusqu'à dix fois meilleure sur la productivité de conception sera nécessaire au cours des dix prochaines années, jusqu'en 2019 afin de maintenir l'effort de conception constante.

1.1 Motivation

En conséquence, il y a deux grands défis dans la conception de la nouvelle génération de MPSOC :

1. Comment améliorer la productivité de la conception afin de réduire le temps de mise en marché (TTM) de système électronique qui est de plus en plus complexe?
2. Comment s'assurer que le projet de conception actuel est adaptable à la technologie de semi-conducteurs qui évolue rapidement?

La productivité de la conception du système sur puce est le défi majeur en matière de technologie de conception. L'écart de productivité de la conception représente le fait que la loi de Moore génère un certain nombre de transistors disponibles, qui croît plus vite que la capacité à les utiliser d'une manière significative. La complexité de silicium et la complexité des systèmes sont à l'origine de cette croissance exponentielle de l'écart de productivité de conception. La complexité de silicium est le résultat des propriétés physiques de la technologie des semi-conducteurs et de l'agrandissement de l'interconnexion globale. Les défis associés à la complexité des systèmes sont la réutilisation, la vérification et le test, ainsi

que l'optimisation de la conception axée sur la rentabilité, la conception de logiciels embarqués, les plateformes d'implémentation fiable, et la gestion des processus de conception.

1.2 Les objectifs de recherche

Des nouvelles méthodologies de conception efficaces sont nécessaires pour surmonter les complexités des systèmes et du silicium afin de remplir les écarts de la productivité de conception croissante. Pour atteindre cet objectif, on propose trois stratégies :

1. la combinaison de divers niveaux de conception du système.
2. la réutilisation des IPs et des composants.
3. l'utilisation des nouvelles technologies.

Dans cette thèse nous présentons une nouvelle méthodologie qui implémente ces 3 stratégies pour résoudre les défis de conception.

Cette nouvelle méthodologie combine ensemble différents niveaux de conception pour prendre les avantages de chaque niveau et surmonter le défaut séparé de chaque niveau. De cette façon, nous pouvons tirer profit des modèles au niveau d'abstraction élevé de la conception de haut niveau du système pour passer au niveau RTL pour alimenter l'information électronique du système autant que possible. Cette méthode garantit aussi que la conception du système est toujours adaptative à la nouvelle technologie de semi-conducteurs. De plus, les développeurs avec des simulateurs de système font face à "un mur de simulation" à cause du temps de simulation des systèmes avec des centaines ou plus de cœurs. L'émulation du CMP de grande taille sur la plate-forme multi-FPGA est une des solutions proposées pour accélérer l'exploration de l'espace de conception des systèmes.

Cette nouvelle méthodologie réutilise les blocs et composants d'IP existants pour accélérer le processus de conception et faciliter la vérification du système. Jusqu'à présent, les IPs réutilisables sont toujours trop élémentaires pour construire rapidement des multiprocesseurs de grande taille. Il est alors nécessaire d'augmenter la taille et la complexité d'IPS que nous appelons le multiprocesseur à petite échelle (small scale multiprocessor SSM) IPs. La conception de multiprocesseurs de grande taille basées sur des SSM IP permet la duplication

et la construction rapide en un temps raisonnable utilisant l'émulation sur multi-FPGA pour une validation et une évaluation rapide de performance.

De nouvelles technologies pour la conception de MPSOC sont appliquées dans cette nouvelle méthodologie :

- le Réseau sur la Puce (NoC), une architecture de communication est utilisée pour régler le problème d'évolutivité du système à grande taille.
- Synthèse de Haut Niveau (HLS), outils de génération de VHDL coprocesseur accélèrent la conception et la réutilisation de la IP.
- La technologie de FPGA reconfigurable embarquée permet la mise en œuvre de NoC reconfigurable sur ASIC.
- la plate-forme multi-FPGA rend l'émulation de système de grande taille réalisable.

La mise en oeuvre complète de ces stratégies et technologies dans notre nouvelle méthodologie constitue une bonne solution pour améliorer la productivité de conception du système et sa fabrication.

2. Définition et état de l'art

Le MPSoC (Multiprocessor System on Chip) est différent du multi-cœur distribué ou le multiprocesseur, parce que tous les éléments de traitement sont intégrés sur une puce. La différence majeure vient de l'architecture de communication. Le multiprocesseur est connecté par un réseau d'interconnexion externe avec une grande bande passante et une latence haute ; tandis que la communication sur puce de MPSoC doit être rapide et la gestion de réseau doit être simple et efficace. Les couches d'intégration à très grande échelle (VLSI) fournissent de nombreux fils pour transférer des signaux de contrôle et des données. La proximité locale des éléments de traitement et des mémoires accélère le transport. Mais le compromis doit toujours être fait entre la surface, la performance et la consommation d'énergie.

Le Réseau sur puce (NoC) est une nouvelle méthode pour les communications au sein de grands systèmes VLSI mis en œuvre sur une seule puce de silicium. Comme la complexité de systèmes intégrés continue à grandir, le NoC fournit la performance améliorée et l'évolutivité en comparaison avec des solutions simples de communication sur puce tels que le point-à-

point et le bus partagé. Avec l'avènement du multiprocesseur de grande taille, le NoC est un choix naturel pour la conception d'architecture. Il peut offrir une séparation entre le calcul et la communication, soutenir la modularité et la réutilisation de la IP via des interfaces standards, gérer des problèmes de synchronisation, servir de plateforme pour le test du système et partant, accroître la productivité de conception.

Le NoC est un réseau de communication qui est utilisé sur une puce. Il est construit de liens multiples de interconnectés par des routeurs. Les données peuvent être transférées de la source à la destination sur plusieurs liens, en faisant la décision de routage sur les routeurs. Un haut niveau de parallélisme est obtenu, parce que tous les liens dans le NoC peuvent fonctionner simultanément sur les différents transferts de données.

Un NoC élémentaire se compose de routeurs, de liens et d'interfaces de réseau. Les routeurs dirigent des données sur plusieurs liens selon le politique de routage. Les connexions logiques des liens sont mentionnées comme la topologie de réseau. L'interface de réseau (l'adaptateur) doit découpler le calcul (les ressources) de la communication (le réseau). Chaque cœur IP est connecté au NoC par une interface de réseau.

Les mesures les plus importantes des NoCs sont la bande passante, la surface de silicium, la consommation d'énergie, et la latence. Tous ceux-ci doivent être réduits au minimum. Une solution de Pareto est prévue pour l'exploration à grande échelle.

Différentes méthodes d'évaluation peuvent être utilisées pour mesurer la performance du système: l'analyse basée sur modèle, la simulation et l'exécution sur des puces réelles. L'analyse mathématique est rapide, mais peu précise. Elle peut être utilisée à la première étape pour la vérification rapide et pour enlever les options inutiles et diminuer l'espace d'exploration. La simulation SystemC est largement utilisée dans la recherche. Lorsque le système devient de plus en plus complexe, la vitesse de simulation n'est plus suffisante. L'émulation sur plate-forme FPGA est proposée comme solution pour l'exploration à grande échelle. Nous soutenons que le temps d'exécution doit être mesuré en temps réel plutôt que le nombre de cycle. La fréquence du système doit être mesurée pour des résultats de simulation.

La communication entre processeurs est importante pour la conception de MPSoC. Le bus est une architecture d'interconnexion traditionnelle pour la conception de système sur puce (SoC). L'AMBA bus d'ARM et le CoreConnect bus d'IBM sont les choix connus pour la conception de processeur commercial. Le bus d'OCP (Open Core Protocol) est proposé

comme un moyen efficace pour simplifier l'interconnexion par la standardisation de protocole d'interface. Comme le nombre de processeur dans des MPSoCs grandit exponentiellement, le réseau sur puce (NoC) est proposé comme la seule solution pour la bande passante de communication requises, l'évolutivité de conception et l'énergie limitée.

Des multiprocesseurs homogènes et hétérogènes sont les deux branches importantes et distinctes de la conception de MPSoC. Les communications inter processeur sont très importantes pour la conception. Jusqu'à présent, il n'y a pas beaucoup de conceptions de MPSoC qui sont basées sur l'architecture de NoC. Une étude montre qu'il n'y a aucune procédure de conception pour les MPSoCs plus grands de 32 cœurs interconnectées par la technologie de NoC.

3. Exploration de conception multi-objectif de NoC au niveau TLM

A la première étape de la thèse, on propose une exploration de l'espace pour la conception multi-objectif de NoC au niveau TLM. L'exploration automatique est nécessaire afin de garantir l'évaluation de toutes les solutions possibles. Bien que certains travaux ont été réalisés dans ce domaine, l'explorations de l'espace de conception proposées sont basées sur différents niveaux d'abstraction. Les modèles de SystemC TLM cachent beaucoup de détails d'implémentation de bas niveau. Il permet une rapide simulation des systèmes complexes au prix de moins de précision. Comment gérer l'exploration de l'espace de conception avec ce manque de précision est l'objectif de ce travail. L'exploration de l'espace de la conception multi-objectif devient un défi parce que la surface et timing du système doivent être extraites des niveaux plus bas de l'abstraction. La représentation au niveau TLM de NoC exige la perspicacité profonde et l'expérience d'implémentation pour interpréter correctement la sémantique correspondante à ce niveau d'abstraction.

La version 2.0 de SystemC TLM a été publiée récemment et elle permet la modélisation du système sur puce au plus haut niveau d'abstraction. Comme le SystemC TLM sont basées sur les transactions, la communication entre les IPs doit être effectuée au niveau des transactions. L'analyse d'architecture peut utiliser les trois styles de codage disponibles qui sont unlimited, loosely-timed et approximately-timed.

3.1 Les modèles de surface

Nous avons utilisé dans cette étude des outils de conception industrielle : NoC Solution de l'Arteris. Il contient deux outils de CAO: NoCexplorer et NoCcompiler, qui se concentrent sur les différents niveaux de simulation. NoCexplorer est un outil de génération et simulation système utilisant le langage SystemC TLM. Les modèles cycle basé de l'Arteris accélère la vitesse de simulation. NoCcompiler peut être utilisé pour générer des codes en VHDL ou SystemC RTL pour NoC.

À ce niveau TLM, la topologie de NoC est représentée par des 'liens' (links). Un lien est caractérisé par son horloge et sa largeur, et il est éventuellement associé à une capacité de mémoire tampon FIFO. Les liens portent les paquets de requête et réponse entre chaque paire de source et destination. N'importe quelle type de topologie, régulière ou irrégulière, peut facilement être décrite utilisant des 'liens'. Pour chaque paire de source et de destination, nous décrivons la séquence des liens comme son routage, dans lequel passe la communication. Pour définir l'architecture du NoC, dans le script de NoCexplorer, on décrit le routage sur un réseau de liens au lieu de routeurs. Le point où se mêle de liaisons représente un routeur au niveau RTL, qui n'est pas un module au niveau TLM. Le NoC et les modèles d'esclave et de maître sont décrits dans un fichier de script comme l'entrée de NoCexplorer.

Après une simulation rapide au niveau SystemC TLM, NoCexplorer donne un rapport de la performance du système. Ensuite, nous pouvons transférer la topologie de NoC à NoCcompiler, pour la simulation et l'implémentation au niveau RTL. Les performances, incluant la latence et la bande passante sont prises comme les fonctions d'objectifs dans l'exploration. La méthodologie de conception décrite a été appliquée à une étude de cas de taille significative.

On construit les modèles de surface au niveau TLM pour les liens et routeurs utilisant le rapport des surfaces au niveau RTL estimée par NoCcompiler. La surface est calculée dans l'unité de porte NAND2. Les modèles de NoC au niveau TLM cachent beaucoup de détails d'implémentation comparés aux modèles correspondants de niveau RTL et ce pour obtenir une simulation rapide. Nous pouvons construire des modèles de surface de niveau TLM pour l'estimation. Le composant de RTL est mis aux mêmes configurations que le modèle TLM correspondant, si ces options sont présentés au niveau TLM. Sinon, les options de composant de RTL restent par défaut. Dans cette étude, la profondeur des liens et le nombre d'ES de

routeur sont modifiées en fonction des différentes configurations de l'exploration. En conséquence, nous changeons la capacité du tampon FIFO du composant et le nombre d'ES pour trouver les relations entre la consommation de surface et ces variables.

Selon les données de l'estimation des surfaces de NoCcompiler, la proportion entre la capacité de FIFO et la consommation de surface est linéaire, qui peut être présenté comme:

$$gates = \begin{cases} 0, depth = 0 \\ 372 * depth - 30, depth \geq 1 \end{cases}$$

où le 'gates' représente la surface de lien et le 'depth' représente la capacité de FIFO de ce lien.

La relation entre la surface et le nombre d'ES du routeur est plus complexe. Grâce à une analyse numérique des données, la relation entre la surface du routeur et son nombre d'ES est:

$$Gates = 72 * X * Y + 273 * Y + 39 * X + 18$$

où le 'gates' représente la surface ; le 'X' représente le nombre d'entrée de ce routeur et le 'Y' représente le nombre de sortie de ce routeur.

3.2 Multi-objective NOC TLM DSE

La procédure de conception, NOCDEX2 est décrite ci-dessous.

NOCDEX2

générer des populations de configuration du NoC aléatoirement par la modification du scénario

```
while (les critères de terminaison non atteint)
  for (toutes les configurations du NoC)
    simuler au niveau TLM et enregistrer les performances
    estimer la surface
    classer toute les configurations
  générer une nouvelle génération de NoC
analyser le front de Pareto final
```

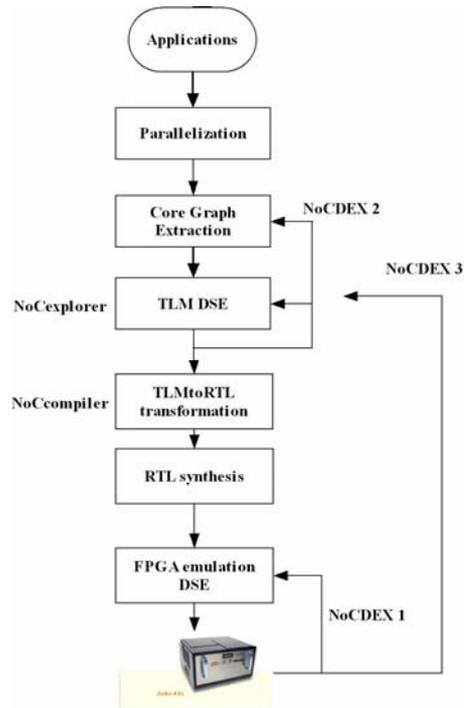


Figure 1 la procédure de conception NoCDEX 3

Combinant NoCDEX 2, notre procédure d'exploration au niveau TLM et NoCDEX, notre procédure d'émulation basée sur la plateforme FPGA, nous proposons une procédure complète de la conception du niveau TLM au RTL, NoCDEX3 qui est montrée dans la figure 1.

Après la parallélisation, le core graphe est extrait d'applications réelles. D'abord NoCDEX 2 est utilisé pour trouver des solutions de Pareto, qui sont utilisées pour l'émulation précise de NoCDEX 1. Si les résultats ne répondent pas à l'objectif de la conception, les résultats sont retournés à la simulation de haut niveau pour la nouvelle parallélisation d'application.

4. NoC reconfigurable sur eFPGA

Le multiprocesseur système sur puce (MPSoC) devrait être utilisé pour des applications multiples qui pourraient présenter des modèles de communication distincts. Comme le nombre d'IP augmente exponentiellement, le problème le plus important de la communication sur puce est de garantir la qualité du service. Le réseau sur puce (NoC) offre une solution éprouvée pour la communication des systèmes sur puce (SoC) complexes. Plusieurs conceptions ont été réalisées. Toutefois, peu d'études ont été faites sur la conception de NoC

pour des demandes d'applications multiples. La conception d'un NoC efficace commun pour ces demandes d'applications multiples pourrait être impossible en raison des exigences divergentes.

Le NoC reconfigurable est une solution potentielle pour ce problème, parce que le réseau est reconfiguré avant l'exécution d'applications afin de répondre aux besoins spécifiques des applications multiples. L'implémentation de cette reconfiguration pourrait être faite en utilisant le circuit d'eFPGA (embedded FPGA). Nous proposons une méthodologie pour spécifier la dimension de secteur d'eFPGA reconfigurable pour NoC. Les résultats d'expérience montrent l'efficacité de notre approche.

L'avantage majeur d'eFPGA est sa capacité de faire des changements après la fabrication du circuit SoC. Sa reconfigurabilité faite sur l'eFPGA est approprié pour des composants sur puce, comme des accélérateurs hardware pour les processeurs afin d'accélérer les applications embarquées, des unités de cryptage des données dans les appareils sans fil qui ont besoin d'être changées de temps en temps, des interfaces d'entrée-sortie pour la transmission de données, les routeurs de NoC qui doivent changer de configuration et routage pour s'adapter aux trafics dynamiques. Les avantages de cette approche permettent d'approvisionner des clients différents avec une seule puce programmable qui peut accommoder des changements des standards ou des spécifications.

4.1 La définition du problème de NoC reconfigurable

Nous supposons comme données du problème que la bibliothèque de fréquence et de surface pour chaque configuration du routeur dans le NoC est disponible. L'objectif est de trouver un NoC hétérogène de haute performance sous contraintes de surface totale d'eFPGAs.

Entrée: (1) un NoC avec N routeurs, (2) la contrainte de surface totale d'un ou plusieurs eFPGAs disponible.

Sortie : le choix de configuration pour chaque routeur dans le NoC et la position de chaque router sur les eFPGA, si plusieurs eFPGAs sont utilisés.

Contraintes: la surface totale des routeurs sur chaque eFPGA ne peut pas dépasser la surface maximum de cet eFPGA où ces routeurs sont placés.

4.2 La solution de programmation linéaire et l'algorithme

Nous allons placer un NoC avec N routeurs sur K eFPGAs. Sur chaque eFPGA, il y a LUT_k luts et RAM_k rams ($k = 1, \dots, K$). Les routeurs peuvent être placés sur n'importe quel eFPGA. Nous devons trouver la configuration et la position de chaque routeur pour maximiser les fréquences du NoC.

Nous introduisons une variable binaire $x_{i,j,k}$ pour représenter le choix de la configuration j et la placement k du routeur i :

$x_{i,j,k} \in \{0,1\}$ pour $i = 1, \dots, N$; $j = 1, \dots, M$ et $k = 1, \dots, K$.

$x_{i,j,k} = 1$, si routeur i est fixé à sa configuration j et il est mis sur le eFPGA k . Sinon, $x_{i,j,k} = 0$

Mettez $lut_{i,j,k}$ et $ram_{i,j,k}$ comme le nombre de luts et rams de switch i , si le routeur est fixé à sa configuration j et il est mis sur le eFPGA k . Et $f_{i,j,k}$ représente la fréquence du routeur i . La formulation ILP de ce problème est la suivante:

$$\text{Max: } \sum_i \sum_j f_{i,j,k} \cdot x_{i,j,k} \quad (1)$$

$$\text{s.t. } \sum_j \sum_k x_{i,j,k} = 1, \forall i \quad (2)$$

$$\sum_i \sum_j lut_{i,j,k} \cdot x_{i,j,k} \leq LUT_k, \forall k \quad (3)$$

$$\sum_i \sum_j ram_{i,j,k} \cdot x_{i,j,k} \leq RAM_k, \forall k \quad (4)$$

L'objectif est de maximiser la fréquence de tous les routeurs dans (1). La contrainte (2) permet de s'assurer que chaque routeur est fixé à une seule de ses configurations et est placé à un seul eFPGA. Dans la contrainte (3), nous nous assurons que les luts totaux de tous les routeurs sur eFPGA k ne dépasseront pas le maximum LUT_k . Et dans la contrainte (4), nous nous assurons que les rams totaux de tous les routeurs sur eFPGA k ne dépasseront pas le maximum RAM_k .

4.3 Algorithme pour NoC reconfigurable sur eFPGAs minimum

Dans le cas de eFPGAs nombreux, il n'est pas nécessaire d'utiliser tous les eFPGA pour le placement du NoC. Donc on cherche à minimiser le nombre d'eFPGAs utilisés. Nous utilisons l'algorithme basé sur la formulation ILP pour résoudre ce problème.

L'idée majeure est de tester la faisabilité du placement des routeurs sur des eFPGAs, d'un eFPGA juste qu'à $K-1$ eFPGAs. Si la valeur de la fonction objectif de la solution n (moins de

K) eFPGAs est égale à la F_{max} maximum de la solution K eFPGAs, et les autres valeurs des fonction objectif des solution m (moins de n) eFPGAs sont inférieures à F_{max} , puis on trouve une solution minimale pour le problème du NoC sur eFPGAs nombreux.

Dans cet algorithme 1, la liste de combinaison des eFPGAs K est construite pour le problème eFPGAs i. Un exemple combinaison de 3 eFPGAs (1,2,3) est:

(1); (2); (3); (1,2); (1,3); (2,3); (1,2,3)

Algorithm 1 NoC on Minimal eFPGAs Algorithm

```
1: calculate ILP value  $F_{max}$  of  $K$  eFPGAs problem
2: build list of all the combination of  $K$  eFPGAs
3: for  $i=1$  to  $K$  do
4:   calculate ILP value  $F_i$  of  $i$  eFPGAs problem
5:   if  $F_i \geq F_{max}$  then
6:     break
7:   end if
```

5. Multiprocesseur de petite échelle (SSM IP)

La future génération de systèmes multiprocesseurs sur puce (MPSoC) sera basée sur des centaines de processeurs connectés par un réseau sur puce (NoC). Un des défis est d'augmenter la productivité de la conception. Nous proposons un multiprocesseur de petite échelle basé sur NoC (SSM IP) comme un élément constitutif des multiprocesseurs à grande taille. Nous décrivons l'architecture d'un tel SSM IP ainsi que les résultats de prototypage sur une seule puce FPGA. Les applications de traitement d'image sont utilisées comme évaluation préliminaire de logiciels parallèles.

5.1 Architecture

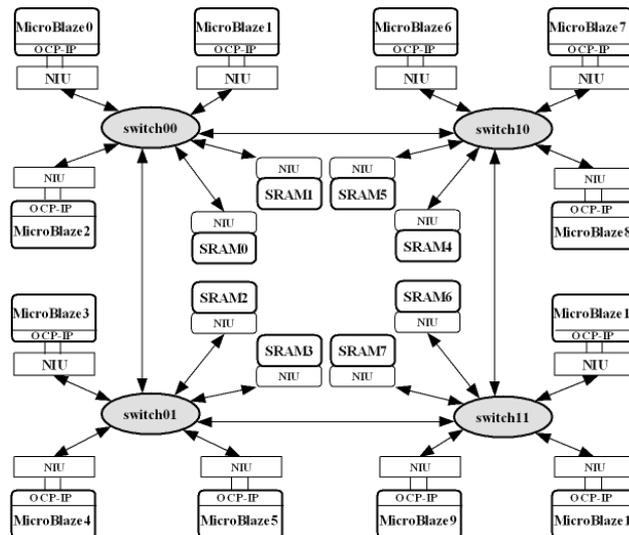


Figure 2 l'architecture du multiprocesseur de petite échelle (SSM IP)

La topologie de notre SSM IP est une maille de groupes (cluster). Le SSM IP est composé de 12 processeurs connectés par une maille 2x2 de routeurs, avec 3 processeurs de MicroBlaze et 2 SRAMs par routeur. La topologie de réseau est maille à cause de ses propriétés d'évolutivité et l'extensibilité. La régularité de mise en page et les retards d'interconnexion sont primordiaux. La topologie maillée fournit des liens courts et est plus facile à placer et router. La performance du dispositif est mieux gérée grâce à des liens courts. Plusieurs projets récents de système sur puce ont utilisé la topologie de maille.

Toutefois, notre architecture d'une maille de clusters est meilleure qu'une maille complète, parce qu'on peut mieux profiter de la localité de données dans un cluster pendant le traitement de l'image pour une application multimédia. Les images peuvent être regroupées également entre les mémoires partagées de chaque cluster, afin que les processeurs appartenant à un cluster puissent traiter la partie d'image associée à ce cluster.

Notre IP SSM est une soft IP. Il est composé des soft IP, qui sont décrites dans le tableau suivant.

Table 1 IPs de SSM multiprocesseur.

Composant d'IP	Description	Source	Version	Nombre
Processeur	Soft IP	Xilinx MicroBlaze	6.00 b	12
Mémoire	Soft IP	Xilinx Coregen 96KB	2.4.	8
Routeur du NoC	Soft IP	VHDL Arteris Danube library	1.10	4

Il y a beaucoup de configurations disponibles pour notre SSM IP. Les 4 exemples se trouvent dans le tableau, qui modifient les configurations de processeur et routeur. Bien que cette exploration de l'espace n'est pas grande, elle illustre les variations de SSM IP. L'application est mis en œuvre sur ces 4 architectures afin de comparer leurs performances.

Table 2 Les 4 versions d'architecture.

	NoC	MicroBlaze
Arch. V1	Fwdpipe	Multiplier
Arch. V2	Fwdpipe+Bwdpipe+Pipe	Multiplier
Arch. V3	Fwdpipe	Multiplier+FPU
Arch. V4	Fwdpipe+Bwdpipe+Pipe	Multiplier+FPU

5.2 Implementation and Results of NL-means filter

L'algorithme de NL-means est programmé en langage C pour chaque processeur dans notre SSM IP. L'image grise de 64x48 est divisée en 12 blocs avec la même taille de dimension de 16x16. Une ligne de 3 blocs est mappée vers un SRAM d'un cluster. Chaque processeur lit un bloc d'image de local SRAM. Après le filtrage de NL-means, les résultats sont envoyés dans un autre SRAM local de ce cluster.

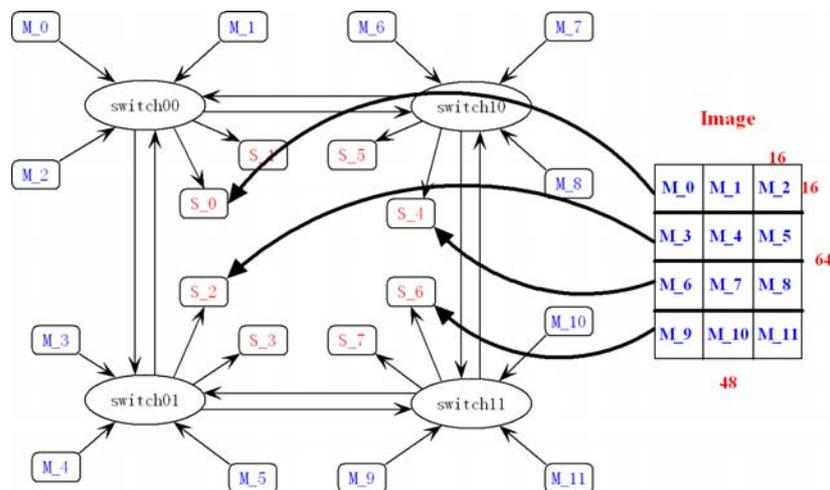


Figure 3 Le mapping d'image aux SRAMs pour l'application de NLMeans filtrage

La Figure 4 montre les performances de l'application NL-means sur les 4 architectures différentes. Afin de mieux analyser la corrélation entre le temps d'exécution et la consommation de hardware, toutes les informations sont fournies dans la même figure. L'axe

Y droit fournit l'information de surface (slice), tandis que l'axe Y gauche fournit l'information de temps d'exécution.

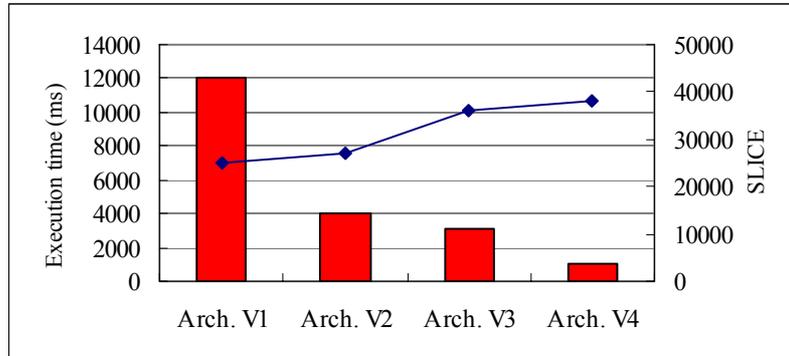


Figure 4 résultats d'implémentation de NLMeans filtrage

L'augmentation de l'utilisation de surface diminue le temps d'exécution dans tous les cas. Toutefois, le gain n'est pas linéaire. Par exemple, entre Arch. V4 et V3 ou entre Arch. V1 et V2, la variation de la surface est moins de 5%, mais le temps d'exécution est réduit de plus de 50%. La fréquence du système est augmentée utilisant plus de pipelines dans le NoC, qui prend peu de slices dans les architecture V2 et V4. Cela souligne l'importance de notre soft SSM, dont les configurations peuvent être changées selon les besoins de l'application. L'unité de traitement flottant (FPU) peut grandement améliorer les performances de calcul du processeur MicroBlaze. La performance du système peut être améliorée par un ordre de grandeur entre les architectures V4 et V1.

6. Multiprocesseur de grande taille (LSM)

La productivité de conception est un des défis les plus importants de la future génération de multiprocesseur sur puce (MPSoC). Nous proposons d'augmenter la productivité de conception en réutilisant notre SSM IP combinée avec les techniques de l'extension rapide. Une implémentation d'un multiprocesseur de 48 coeurs sur une plateforme de 4 grand FPGA a validé notre approche.

6.1 Extension du SSM IP au LSM

Afin d'atteindre une productivité de conception rapide de MPSoC de grande taille, nous avons besoin: (1) de réutiliser notre multiprocesseur à petite échelle (SSM IP) et d'ajuster

automatiquement les configurations du NoC (2) d'intégrer les outils EDA industrielles dans la procédure de conception. En raison de sa grande taille et temps de simulation prohibitifs au niveau RTL, nous avons besoin d'émulation en place de simulation des performances de multiprocesseur de grande taille.

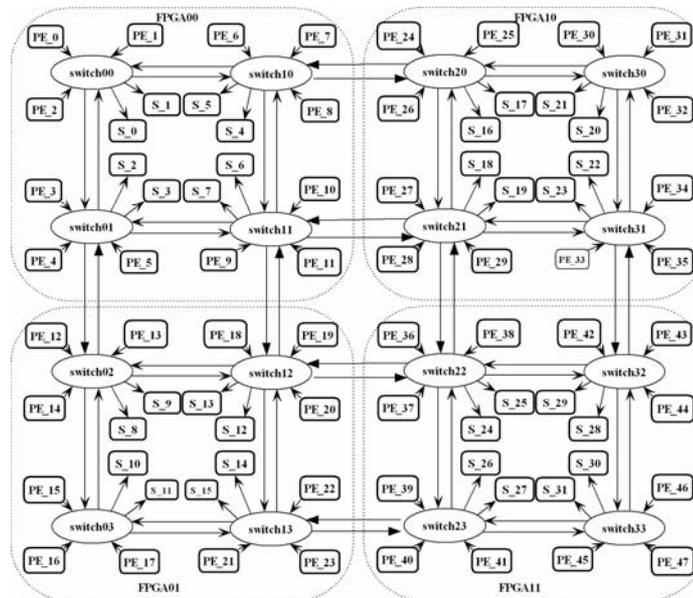


Figure 5 l'architecture du multiprocesseur de 48 coeurs

Un multiprocesseur de 48 coeurs est implémenté sur la plate-forme ZEBU-UF 4. Notre SSM IP est réutilisée pour accélérer la conception de multiprocesseur. Tous les fichiers de configuration de SSM IP sont réutilisées pour l'extension, qui peuvent largement réduire le temps de synthèse de ce multiprocesseur. En double sur 4 FPGA, tous les composants SSM ne seront pas changés, sauf le NoC adapté pour la nouvelle 4x4 topologie maillée.

6.2 Intégration des outils CAO et la procédure de conception

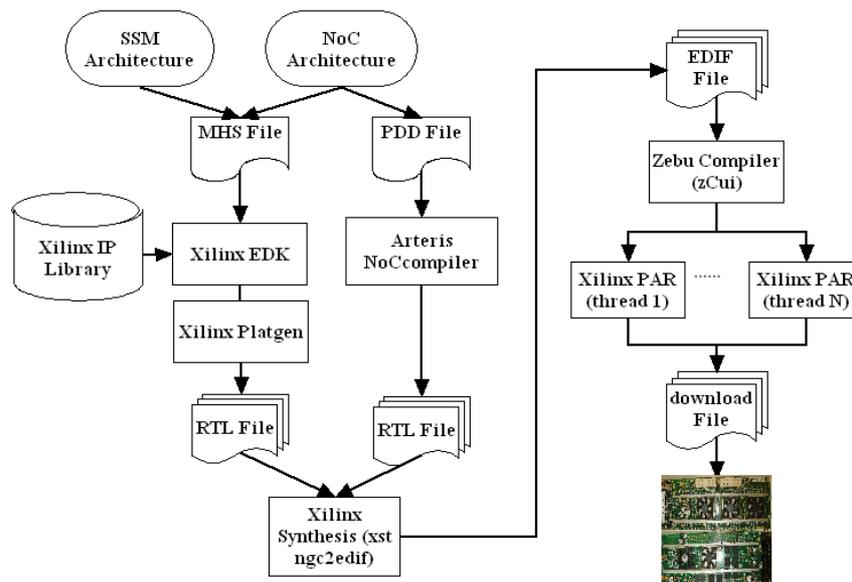


Figure 6 La procédure de conception pour MPSoC sur la plateforme multi-FPGA

Les outils CAO de trois sociétés commerciales sont mis en semble pour générer notre MPSoC de 48 coeurs. L'outil EDK de Xilinx est utilisé pour générer nos SSM multiprocesseurs. Une fois que les fichiers de RTL de SSM sont produits, ils sont réutilisés pour la synthèse de multiprocesseur de grande taille, ce qui peut largement réduire le temps de conception du système. Différents fichiers de NoC sont synthétisés pour chaque SSM sur les différentes puces FPGA de la plateforme en changeant le tableau de routage de chaque routeur selon le politique de routage. Ces fichiers RTL des NoCs sont générés par NoCcompiler, outil d'Arteris. Le compilateur ZEBU d'EVE prend les fichiers EDIF convertis par des outils de synthèse de Xilinx pour l'implémentation sur FPGA. Enfin l'outil de placement et routage est utilisé pour générer les fichiers de téléchargement au FPGA. Cette phase peut être parallélisée afin de réduire le temps de conception. Les résultats de surface et performance sont obtenus par l'émulation sur la plateforme multi-FPGA.

6.3 Modèles de parallélisme et l'implémentation

Une application typique de traitement du signal peut être divisée en plusieurs blocs de fonction et parallélisée par le mapping des blocs sur des différents éléments de traitement. Ces éléments de traitement peuvent travailler en pipeline pour la parallélisation de tâche.

6.3.1 Modèle de Fork-Join

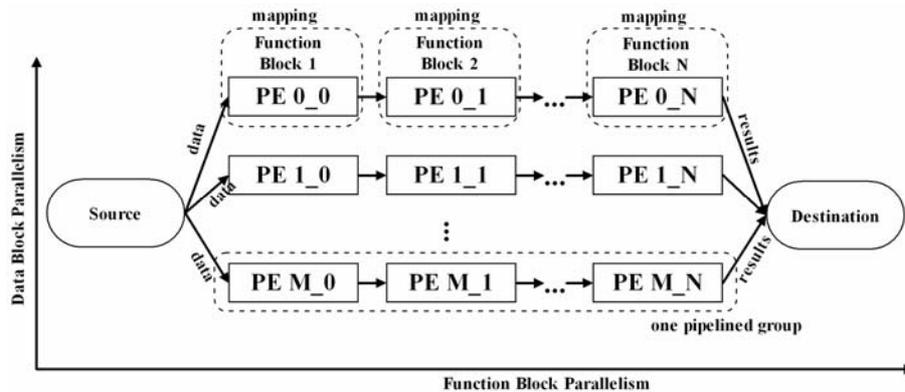


Figure 7 Modèle de Fork-Join avec les parallélismes de donnée et tâche

Deux types des parallélismes, le parallélisme de donnée et le parallélisme de tâche, sont combinés ensemble pour atteindre une meilleure performance. Les deux parallélismes sont réunis pour travailler en modèle de Fork-Join, montré dans la Figure 7.12. Dans le parallélisme de donnée, nous distribuons différents blocs de données aux groupes de traitement différents. Ces groupes travaillent sur les données reçues en parallèle. Dans le parallélisme de tâche, toutes les fonctions d'application sont divisées en blocs et placées sur des PEs (Processor Element) séquentiellement. Chaque PE obtient des données d'entrée, calcule les blocs de fonction associés et envoie les résultats au processeur PE suivants et enfin à la mémoire de destination. Les PEs mappés avec des blocs de fonction travaillent ensemble comme un groupe en pipeline.

6.3.2 L'implémentation de parallélisme sur MPSoC basé sur NoC

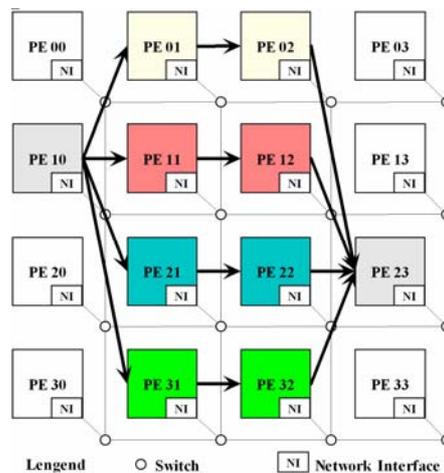


Figure 8 Implémentation de modèle de Fork-Join sur MPSoC avec NoC

Le MPSoC basé sur NoC est la tendance du future multiprocesseur à grande taille en raison de la flexibilité du NoC. Pour paralléliser l'application, plusieurs PEs dans le MPSoC sont divisés en groupes distincts pour la parallélisation de donnée. Afin de minimiser les temps de latence de la communication, les PE dans le même groupe doivent être aussi proches que possible. L'architecture simplifiée d'un MPSoC et une implémentation de notre modèle de Fork-Join sont illustrés dans la figure. Dans cet exemple, PE10 travaille comme source et envoie des blocs de données distincts à 4 groupes différents de PE en des couleurs différentes dans la figure. L'application est divisée en 2 blocs de fonction et ils sont placés sur les deux PEs dans chaque groupe. Enfin chaque groupe envoie les résultats vers PE23 qui sert comme destination. Il y a au total $4 * 2 = 8$ PEs utilisés dans cet exemple d'application.

6.4 L'implémentation des Parallélismes sur MPSoC

Le code C séquentiel de cryptage TDES est composé d'une permutation Forward (FO), 48 appels à une macro F et enfin une permutation inverse (IP). Pour utiliser pleinement notre multiprocesseur, le parallélisme de données et le parallélisme de tâche sont combinés pour obtenir une meilleure performance. Un exemple est donné dans la Figure 7.19:

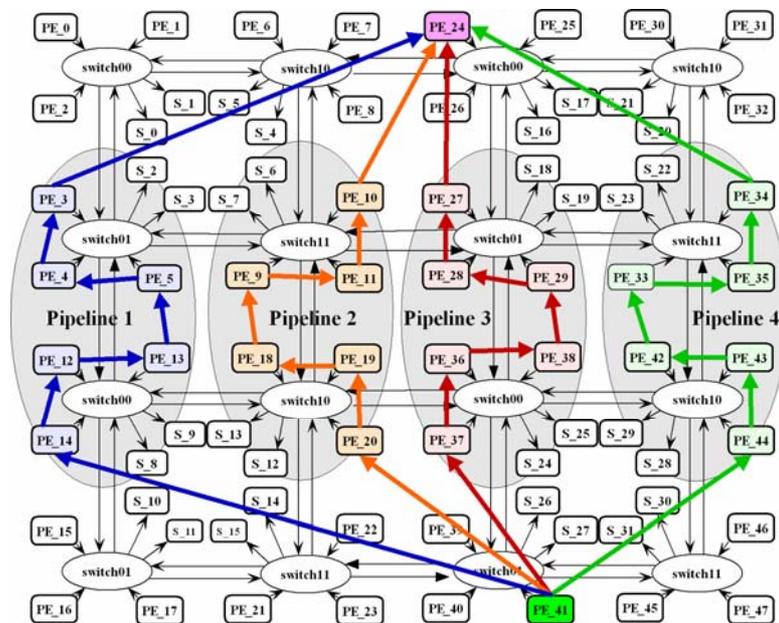


Figure 9 Un exemple de mapping de modèle Fork-Join sur 48-PE multiprocesseur

1. 24 PEs de MicroBlaze sont choisis pour l'implémentation;
2. Toutes les données sont divisées en 4 blocs et les 24 PEs sont divisés en 4 groupes;
3. Pour crypter chaque bloc de données, chaque groupe dispose de $24 / 4 = 6$ PEs;
4. Dans chaque groupe en pipeline, chaque PE MicroBlaze calculera $48 / 6 = 8$ appels de macro F.

L'exploration aide à trouver un bon compromis entre le Parallélisme des tâches et le parallélisme de données. Dans cet exemple, au maximum 24 PEs sont utilisés, la combinaison du parallélisme de données et le parallélisme des tâches est répertorié dans la Tableau 3:

Table 3 combinaison des parallélismes de donnée et tâche

Nombre de group en pipeline	Nombre de PE dans un groupe	Nombre de micro associé a un PE
24	1	48
12	2	24
8	3	16
6	4	12
4	6	8
3	8	6
2	12	4
1	24	2

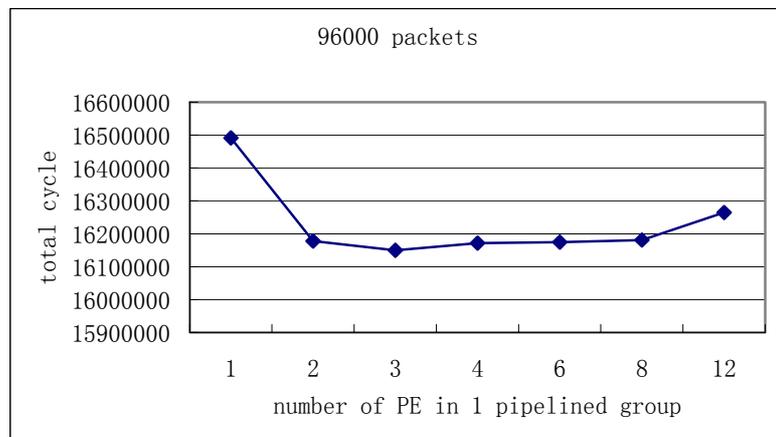


Figure 10 temps d'exécution de 96000 paquets

Dans la Figure 10, il est clair que le temps d'exécution est le plus court dans la combinaison de 8 groupes de PE en pipeline, chacun avec 3 processeurs MicroBlaze dans le groupe. C'est donc le meilleur compromis entre le parallélisme de données et le parallélisme des tâches. Et ce résultat montre l'impact de l'architecture du système sur les performances des applications parallèles.

7. Conclusion

Dans cette thèse nous avons présenté une nouvelle méthodologie de conception de MPSoC à grande taille pour résoudre le défi de la complexité de conception et augmenter la productivité. Pour atteindre cet objectif, on propose et utilise trois stratégies:

1. la combinaison de divers niveaux de conception du système : du niveau TLM à l'émulation de FPGA
2. la réutilisation des IPs et des composants : extension de SSM IP à 48-coeur et 672-coeur MPSoC
3. l'utilisation des nouvelles technologies : Arteris NoC, M2000 eFPGA, EVE plateforme multi-FPGA

Une étude des travaux récents sur la conception de NoC et de MPSOC montre qu'il n'y a aucune procédure de conception mûre pour la future génération de MPSoC à grande taille. L'architecture d'intercommunication de MPSOC affecte énormément la performance du système. Trois méthodes d'intercommunication différentes ont été utilisées : bus, crossbar et réseau sur puce (NoC). Et le NoC est proposé comme la seule solution d'intercommunication pour le futur MPSoC à grande taille. Les outils d'Arteris pour la conception de NoC sont utilisés comme le soutien industriel pour notre procédure de conception.

La contrainte de temps réel doit être prise en compte pendant la conception de MPSOC. Des multiprocesseurs homogènes et hétérogènes sont les deux branches importantes et distinctes de la conception de MPSOC. L'analyse et la comparaison des méthodologies différentes pour la conception de MPSOC aident à mieux les comprendre et à surmonter leurs défauts. Le mapping de "core graphe" aux topologies de NoC est connu comme NP-Hard. Des algorithmes heuristiques sont les seules solutions pour obtenir le résultat proche à l'optimal selon des fonctions objectif différentes. Différents algorithmes d'approximation sont proposés pour réduire le temps d'exécution de la programmation linéaire en nombres entiers.

L'exploration de l'espace de conception du réseau sur puce peut être faite à plusieurs niveaux d'abstraction, de la transaction juste qu'à l'émulation. D'abord on propose un procédure de

conception entièrement automatique pour le réseau sur puce au niveau de TLM. La combinaison de cette procédure avec notre travail d'émulation suivant permettra d'obtenir des solutions satisfaisantes.

Le réseau sur puce reconfigurable exige le support de hardware reconfigurable efficace dans l'environnement d'ASIC. L'apparition d'eFPGA IPs permet l'intégration de secteur reconfigurable dans des puces d'ASIC. L'organisation et le dimensionnement de cette zone sont des questions importantes à traiter pour maximiser l'efficacité du réseau sur puce reconfigurable. Notre méthodologie de programmation linéaire y apporte une solution.

La prochaine génération de MPSoC sera basée sur des centaines de processeurs. la conception de MPSoC est très complexe. Pour accomplir le travail en un temps raisonnable, nous proposons un SSM IP comme un composant élémentaire pour la conception de multiprocesseur à grande taille. Ce SSM IP est basé sur un NoC de topologie Cluster-Mesh. Il a été entièrement évalué sur une grande plateforme FPGA. La conception de multiprocesseurs à grande taille utilisant notre SSM IP est très rapide. L'effort de conception principal est la connexion entre IPs et l'adaptation d'adressage de NOC.

Nous avons validé notre approche sur un multiprocesseur de 48 coeurs en étendant automatiquement notre SSM IP de 12 coeurs et nous avons étendu finalement à un MPSoC de 672 coeurs sur la plate-forme multi-FPGA Zebu-XXL d'EVE. Différents outils de conception industrielle ont été mis ensemble dans notre procédure de conception automatique.

Une procédure de conception automatique pour des données parallèles et des traitements en pipeline est proposée pour l'application au traitement du signal sur le multiprocesseur avec NoC, utilisant l'application cryptographique au TDES (Triple Data Encryption Standard) comme un exemple. Notre procédure explore par l'exécution sur la plate-forme d'émulation multi-FPGA pour la mise en oeuvre de logiciel parallèle avec l'exploration de placement de tâche et l'analyse de granularité de tâche. Un moniteur hardware conduit le processus de placement de tâche pour réduire les congestions de la communication. Dans la deuxième phase, des accélérateurs hardware générés par la synthèse de haut niveau sont ajoutés à notre

procédure pour explorer les compromis entre la performance et la surface en privilégiant toujours la base de multiprocesseur.

L'association de l'OCP-IP a proposé une suite de micro-benchmark pour réseau sur puce. Nous avons évalué ces benchmarks par l'exécution réelle sur nos multiprocesseurs de grande taille avec NoC.

7.1 Travaux Futurs

La conception de MPSoC à grande taille est un secteur de recherche nouveau et ouvert. Dans la thèse, on a proposé des approches qui peuvent être étendues dans plusieurs directions de recherche différentes.

La consommation d'énergie est très importante pour le système électronique, non seulement pour la raison commerciale, mais aussi pour la protection de l'environnement. La bibliothèque de modèle de consommation d'énergie peut être intégrée dans notre procédure de conception pour explorer les compromis entre la performance, la surface et la consommation d'énergie. La combinaison de cette procédure avec nos travaux précédents au niveau RTL permettra une solution complète.

L'exploration de MPSoC à grande taille est très complexe et prend beaucoup de temps. Profitant de la combinaison de différents niveaux, de TLM à RTL, l'espace d'exploration est diminuée à haut niveau pour accélérer l'exploration. La théorie de réseau de neurone et la technologie PR de FPGA peuvent aussi aider à atteindre ce but.

Les travaux futurs consistent à ajouter la gestion de reconfiguration dans la zone d'eFPGA reconfigurable pour des accélérateurs hardware ainsi que la parallélisation automatique de logiciel.

7.2 Contribution

Notre contribution principale est la procédure automatique pour la conception de MPSoC à grande taille basée sur la réutilisation de SSM IP. Basée sur ce principe, une procédure de conception automatique pour des données parallèles et des traitements en pipeline est proposée pour l'application au traitement du signal sur le multiprocesseur avec NoC, utilisant