



# Traitement d'images sur la machine ArMen

P.Besnier J.Champeau M.Daoud B.Pottier  
LIBr-UBO, Dépt d'Informatique  
UFR Sciences, BP 452, 29275 Brest  
e-mail: champeau@univ-brest.fr

## RÉSUMÉ

Dans le cadre d'un projet axé sur une architecture parallèle, la description de la machine ArMen et l'implantation des opérateurs de traitement d'images de bas niveau sont présentées. La machine ArMen est une machine parallèle MIMD, associée à un réseau de FPGAs interconnectés. Cette couche logique reconfigurable permet de synthétiser des services de contrôle qui s'exécutent en recouvrement de l'application parallèle et des services de calcul à grain fin qui ne sont pas résolus efficacement dans les machines MIMD classiques. La croix d'Abingdon est utilisée comme support pour exposer la méthode d'implantation des opérateurs de bas niveau sur une telle machine.

## ABSTRACT

As part of a parallel architecture project, the low level image processing operator implementation and the description of the ArMen machine are present. The ArMen machine is a MIMD parallel machine joined to an interconnected FPGA network. This programmable logic layer allow to synthesized control services which are executing in total overlapping of the parallel application and fin grain parallel computing services which are not solved efficiency in classical MIMD machines. The benchmark of the Abingdon cross is used as a support to show the implementation method of the low level operators on such machine.

## Introduction

Les opérations de bas niveau en traitement d'images nécessitent de nombreux calculs à grain fin sur de larges tableaux de données. Les machines parallèles MIMD sont efficaces pour les phases d'analyse et de reconnaissance de formes mais sont mal adaptées aux traitements de bas niveau. L'architecture de la machine ArMen, exploitant en mode SPMD des circuits reconfigurables, apporte des solutions pour les phases de calcul intensif à grain fin. Après une description de la machine ArMen et de la méthode de synthèse des circuits, la démarche et l'implantation d'opérateurs sont exposés en utilisant une séquence de traitement sur la croix d'Abingdon.

## 1 La machine ArMen et ses Outils

La machine ArMen est une architecture parallèle MIMD intégrant une couche logique reconfigurable[4]. Chaque nœud comporte un processeur, actuellement de type Transputer, une mémoire locale de 1 à 4 Méga et un circuit reconfigurable, FPGA de Xilinx. La réalisation actuelle est faite sur des cartes au format double europe, disposées dans un bac. Une station de travail, contenant une carte avec transputer, autorise un accès à la machine ArMen via les liens de transputer.

Chaque processeur configure localement son FPGA en 100ms, pour le XC-3090 utilisé actuellement. Ce temps de configuration tend à ralentir une application demandant des reconfigurations dynamiques



mais la technologie des circuits reconfigurables suit la même évolution que celle des autres composants, et déjà une récente série permet de diviser par 10 le temps de configuration. Sur un nœud, les FPGAs disposent de 4 ports de 32 bits, le port *nord* est situé directement sur le bus du processeur et les ports latéraux permettent de connecter les FPGAs entre eux pour former une ressource d'intégration partagée par tous les nœuds. Le port *sud*, actuellement libre, peut réaliser un interface souple et rapide pour un périphérique.

La machine MIMD possède son propre système de communication qui est supporté en l'occurrence par les liens de Transputer. Les FPGAs, connectés entre eux, forment la couche logique reconfigurable qui se présente comme une alternative pour des communications rapides et des calculs à grain fin. Cette ressource logique synthétise des services pour la machine MIMD bâtis sur plusieurs modèles architecturaux :

- **les opérateurs locaux** sont des coprocesseurs dédiés, disponibles uniquement pour le processeur local, avec comme exemple la réalisation d'un filtre  $\Sigma$  pour des images codées sur 8 bits [6],
- **les opérateurs globaux** sont synthétisés dans la totalité de la couche logique. Les FPGAs sont connectés en anneau et résolvent les dépendances de données par des communications locales très rapides. Ces opérateurs traitent avec efficacité de larges tableaux de données de taille non standards, 1-2-4 bits, soumis à des traitements réguliers sur tout l'espace de données. Ils sont alimentés localement par chaque processeur,
- **les processeurs globaux** mettent en œuvre des services de contrôle pour les applications distribuées exécutées sur la machine MIMD[3]. Ils permettent par exemple la détection de condition globale stable ou d'assurer une exclusion mutuelle. Ces algorithmes de contrôle, implantés sous forme de pipeline horizontaux, parcourent toute la couche reconfigurable et s'exécutent en total recouvrement de l'application.

Les opérateurs globaux sont utilisés pour les algorithmes de traitement d'images de bas niveau, leur implantation et leur méthode de synthèse sont détaillées. Sur la machine ArMen, l'espace de données est réparti par bandes sur les nœuds. Les dépendances de données locales sont intégrées par communication entre les circuits, ce qui permet de traiter toute une ligne, en un seul cycle écriture-lecture. Si l'espace de données est plus important que la taille de la machine, une mise à l'échelle est effectuée en attribuant à un nœud la gestion des dépendances entre bandes. Ce nœud spécial, nommé nœud de marges, permet d'assurer la cohérence pour un grand nombre d'itérations sans communications entre processeurs.

Une bonne efficacité sur des calculs réguliers est obtenue en exploitant l'opérateur global en mode SPMD[2]. La couche reconfigurable assure la synchronisation des processeurs lors de leur accès à l'opérateur. De même, la détection de l'événement, "*début de traitement*", est faite avec un processeur global synthétisé dans les mêmes circuits que l'opérateur global.

Le modèle d'exécution des opérateurs globaux est basé sur l'application d'une fonction sur l'ensemble des données selon un mode parallèle et synchrone. Une spécification à partir des automates cellulaires définit le comportement de ces opérateurs réguliers, exhibant un parallélisme à grain fin.

Les automates cellulaires s'expriment à l'aide d'une fonction de transition opérant sur le voisinage d'une cellule centrale. Pour synthétiser les circuits de la couche reconfigurable, un langage, CCEL, a été développé pour faciliter la spécification des automates cellulaires. Deux sections sont présentes dans ce langage, une déclare la structure du voisinage et des cellules, l'autre exprime la fonction de transition en langage C. Son compilateur génère la partie applicative des circuits en se reposant sur une architecture prédéfinie, un tableau linéaire de fonctions alimenté par un registre à décalage. Chaque FPGA mémorise toutes les données de la direction nord sud dans le registre et communique avec les circuits voisins pour obtenir toutes les données absentes dans la direction est ouest. Le compilateur fournit une description booléenne du circuit qui est répartie dans les blocs logiques des FPGAs par un outil de synthèse. La phase de routage est effectuée par la chaîne de développement de Xilinx. Les temps de toute la chaîne, compilation et placement/routage, varient de 1 heure à 3 heures suivant la complexité de la fonction de transi-

tion.

## 2 Application au traitement d'images

Les automates cellulaires à deux dimensions permettent de spécifier et de mettre en œuvre les opérateurs de morphologie mathématique développés dans [7]. L'élément structurant correspond à la notion de voisinage de l'automate cellulaire et la transformation morphologique à la fonction de transition. Dans une première étape, les primitives de base, érosion et dilatation sur des images binaires, ont été implantées. La fonction de transition est un *ET* logique pour l'érosion et un *OU* logique pour la dilatation. Le voisinage choisi pour ces opérations est de taille (3 \* 3). A partir de ces deux primitives, des opérateurs morphologiques sont constitués par simple combinaison. Par exemple, l'ouverture (érosion suivie d'une dilatation) et la fermeture (dilatation suivie d'une érosion) sont utilisées comme opérateurs de lissage.

Les fonctions de transition de ces opérateurs n'utilisent pas beaucoup de ressource logique dans les circuits ce qui permet de synthétiser plusieurs opérateurs par FPGAs. Ils sont alors, soit exploités en pipeline, plusieurs itérations sont effectuées en un seul cycle écriture-lecture, soit sélectionnés par le processeur avec un décodage d'adresse implanté dans les circuits.

Dans une seconde étape, le benchmark de la croix d'Abingdon[4] a été utilisé pour l'évaluation des performances[1] et la description de la démarche à mettre en œuvre sur une séquence de traitement de bas niveau sur la machine ArMen. Le but du benchmark est d'extraire le squelette d'une croix, d'une largeur de 64 pixels, située au centre d'une image de taille (512 \* 512) bruitée par un bruit gaussien.

La réalisation a été effectuée sur une des séquences de traitement proposée dans l'article de référence [4]. Elle se compose des opérateurs suivants :

- *Le seuillage* réalise la binarisation de l'image suivant un seuil fixé. Cet opérateur traite l'image pixel par pixel, il a donc 8 bits en entrée et un seul en sortie. Ainsi l'implantation de quatre opérateurs par circuit permet de traiter 4 pixels en parallèle. Cette opération, sans dépendance de données, est effectuée par un *opérateur local* sur chaque nœud.

- *Le filtrage médian* opérant sur différents voisinages, horizontal, vertical, et carré (3\*3), élimine les pixels parasites. Chaque filtre agit sur un voisinage différent ce qui implique la spécification et la réalisation d'un *opérateur global* par filtre pour cette phase.
- *La squelettisation* est un amincissement itéré jusqu'à stabilité de l'image. A la fin de ce traitement, l'épaisseur de la croix est d'un pixel. L'algorithme de squelettisation utilisé, a été défini par Stefanelli et Rosenfeld dans [8]. Il opère sur un voisinage de base (3 \* 3) étendu et il présente l'avantage d'être expliciter avec une seule fonction de transition. Parmi plusieurs algorithmes, celui-ci est le plus adapté à l'architecture car un seul traitement est itéré donc un seul opérateur global est nécessaire.

Les opérateurs de filtrage sont appliqués successivement sur l'image afin d'obtenir une meilleure image de départ pour l'algorithme de squelettisation. Cette chaîne de traitement montre l'utilisation de la couche reconfigurable pour une application entière. La première opération utilise les FPGAs pour la synthèse d'un coprocesseur dédié au seuillage, puis les traitements suivants sont effectués sur la base des opérateurs globaux. L'exploitation du parallélisme à grain fin de ces opérateurs permet d'obtenir de bonnes performances sur les images binarisées. Les temps de traitement, en millisecondes, des différentes phases sont indiqués figure 1.

Opérateurs	Nœuds de traitement			
	1	6	14	30
Seuillage	89.8	15	6.4	3
Filtrage H	179.6	29.9	12.8	6
Filtrage V	179.6	29.9	12.8	6
Reconfiguration	100	100	100	100
Filtrage 3 * 3	179.6	29.9	12.8	6
Squelettisation	5746.2	957.7	410.4	191.5
Total	6474.8	1162.4	555.2	312.5

Figure 1 : Performances pour la séquence de traitement en ms, H=Horizontal et V=Vertical.

Dans la séquence de traitement, une reconfiguration est présente car le nombre de blocs logiques utilisés par les trois premiers opérateurs approche



les deux tiers de la capacité d'un circuit, ce qui est proche de la limite en nombre de blocs pour obtenir un circuit routé. La squelettisation possède un temps de traitement plus élevé car un plus grand nombre d'itérations est nécessaire. De plus, seulement 8 bits sont introduits en parallèle dans chaque circuit car la fonction est plus importante en nombre de blocs logiques et de variables manipulées en entrée donc les 32 fonctions n'ont pas pu être routées. Avec ces performances, la machine ArMen à 32 nœuds se situe au milieu des machines spécialisées pour le rapport qualité, prix et performances. Celles-ci sont à moduler bien entendu en fonction du nombre de nœuds de la machine, et surtout en tenant compte du temps de cycle, écriture-lecture, du transputer qui est de  $1,4\mu\text{s}$ . Le temps d'accès du transputer est supérieur au temps de calcul des opérateurs dans les FPGAs. Les processeurs actuels, comme les sparcs, exécutent la même opération en  $0,2\mu\text{s}$ .

### 3 Conclusion et perspectives

Dans une application de traitement d'images, la machine ArMen utilise toutes les capacités de la couche reconfigurable, les opérations locales à un nœud ou les opérations requérant la participation de tous les nœuds. Cette démarche suivie pour la croix d'Abingdon peut être appliquée à d'autres cas. En effet, les opérateurs locaux sont utilisés en première passe pour préparer les données pour permettre l'utilisation des opérateurs globaux. Ensuite les traitements sont enchainés en profitant des communications rapides internes à la couche logique.

Les résultats obtenus sont bons pour une machine parallèle MIMD qui reste *généraliste* mais qui peut temporairement se spécialiser pour effectuer des traitements intensifs en mode SPMD.

Pour cette application, l'architecture et la mise en œuvre peuvent tendre vers une implantation plus *industrielle* en tenant compte des points suivants :

- une répartition des opérations sur différents nœuds permet d'exécuter la séquence en pipeline,
- la série 4000 des FPGAs de Xilinx augmente la capacité des circuits, ce qui permet d'intégrer un plus grand nombre d'opérateurs dans un seul circuit,
- si le Transputer T800 est remplacé par un processeur actuel, un gain supplémentaire

est attendu.

Outre les améliorations architecturales, l'amélioration de la méthode de synthèse des circuits peut permettre de traiter plus de bits en parallèle dans le cas des fonctions occupant un grand nombre de blocs logiques. Parmi les autres axes du projet, des compilateurs sont développés pour les processeurs globaux à partir de formalismes de haut niveau, UNITY et CSP. Il est nécessaire d'avoir un formalisme commun pour exprimer les opérateurs et les processeurs globaux de la couche reconfigurable.

### Bibliographie

- [1] P. Besnier. Projet armen, benchmark de la croix d'abingdon. Technical Report Rapport de projet de fin d'étude, ENIB, LIBr-UBO, June 1992.
- [2] J. Champeau, B. Pottier, and S. Rubini. Implantation et exploitation d'opérateurs partagés dans un réseau de processeurs MIMD. In *4<sup>es</sup> Rencontres du Parallélisme*, Villeneuve d'Ascq, March 1992.
- [3] J.M. Filloque, E. Gautrin, and B. Pottier. Efficient global computation on a processor network with programmable logic. In *Proceedings of PARLE'91*, number 505 in LNCS, pages 55–63, Eindhoven, NL, June 1991. SV.
- [4] K. Preston Jr. Evaluation of multicomputers for image processing. Academic Press, 1986.
- [5] Bernard Pottier. *ArMen, une machine parallèle intégrant un réseau de circuits logiques reconfigurables*. PhD thesis, Université de Rennes I, France, June 1991.
- [6] Bernard Pottier and Dominique Lavenier. High rate sigma filtering, feasibility studies on processors networks. In *Proceedings of IFIP Workshop on Parallel Architectures on Silicon*, pages 182–196, Grenoble, France, November 1989.
- [7] J. Serra. *Image analysis and mathematical morphology*. Academic Press, 1982.
- [8] R. Stefanelli and A. Rosenfeld. Somme parallèle thinning algorithms for digital pictures. In *JACM*, 1971.