Reconfigurabilité des FPGAs en traitement d'image : contraintes sur l'architecture et critère d'évaluation des performances

Gilles MILLON, Michel ROUSSEL

LAM - Groupe de Traitement d'Image
IUT de Troyes
9 rue de Québec, BP 396, 10026 Troyes Cedex
g.millon@iut-troyes.univ-reims.fr, m.roussel@iut-troyes.univ-reims.fr

Résumé – Cet article présente l'exploitation dynamique de la reconfigurabilité des FPGAs dans le domaine du traitement d'image bas niveau en temps réel. Nous y relevons les différentes contraintes qu'une architecture matérielle doit respecter pour travailler dans ce cadre. Nous proposons une architecture et sa validation par l'implantation d'une chaîne de traitement d'image. Enfin, nous proposons un critère de mesure de l'opportunité de la mise en œuvre de la reconfigurabilité dynamique des FPGAs.

Abstract – This article presents the exploitation of the Run Time Reconfiguration of FPGAs for real time low level image processing. We list the different constraints, the hardware architecture has to satisfy in order to exploit the Run Time Reconfiguration. We make a proposal of a hardware architecture model and present an image processing chain implantation to validate our proposal. Finally, we propose a mesure of the opportunity to use the Run Time Reconfiguration of FPGAs.

1. Introduction

Les performances des systèmes de calcul à base d'ASICs se paient par l'exclusivité de leur utilisation et par un coût de développement élevé. Pour satisfaire des critères économiques, des systèmes à microprocesseur d'usage général sont conçus. Leur flexibilité, liée à leur aspect programmable, les autorise à traiter une grande diversité d'applications mais leurs performances sont limitées. En outre, leur production en très grande série les rend très bon marché. L'idéal serait de combiner la puissance de calcul des systèmes matériels, que sont les ASICs, à la flexibilité des systèmes programmables que sont les microprocesseurs d'usage général.

Dans cette optique, les FPGAs SRAM, par leur reconfigurabilité rapide, in situ et illimitée, offrent des perspectives que nous nous proposons d'étudier dans le domaine du traitement d'image bas niveau (TIBN). Il s'agit d'exploiter dynamiquement la reconfigurabilité du système pour la mise en œuvre d'une même application.

2. Etat de l'art

Des études ont été menées pour exploiter la reconfigurabilité des FPGAs de manière :

- à réduire les ressources matérielles nécessaires à l'implantation d'une application ;
- à accélérer les calculs et à accroître la flexibilité du système de calcul de manière à le réutiliser le plus possible.

Cette exploitation de la reconfigurabilité, plus ou moins sophistiquée, dépend des caractéristiques technologiques des FPGAs mais aussi de l'esprit dans lequel elle est mise en œuvre. Dans tous les cas, une démarche d'Adéquation Algorithme Architecture est cruciale.

2.1 Définitions

La ***reconfigurabilité statique*** consiste à donner une seule configuration au système pour répondre à tous les besoins d'une application (figure 1). L'architecture matérielle doit être suffisamment souple pour que diverses applications puissent être implantées. La contrainte sur les circuits FPGAs est relativement légère puisqu'il suffit qu'ils présentent une reconfigurabilité illimitée et in situ.

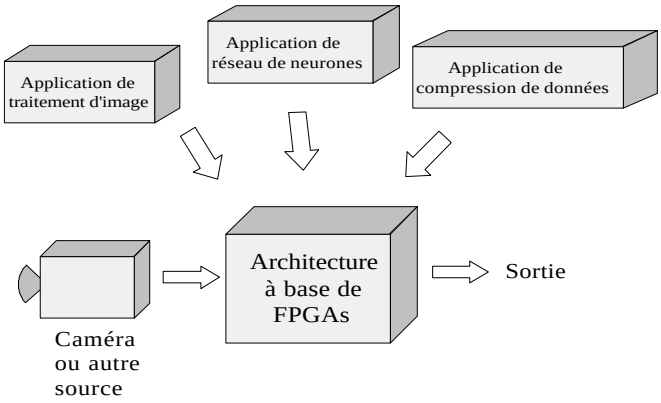

Fig. 1: Principe de la reconfiguration statique.

Pour la ***reconfigurabilité dynamique globale***, il s'agit, à présent, d'exploiter la reconfigurabilité du système au sein d'une même application (figure 2). La mise en œuvre passe par le partitionnement de l'application en plusieurs étapes temporellement indépendantes puis par les configurations successives de celles-ci. La reconfigurabilité est qualifiée de globale dans la mesure où les circuits sont totalement reconfigurés.

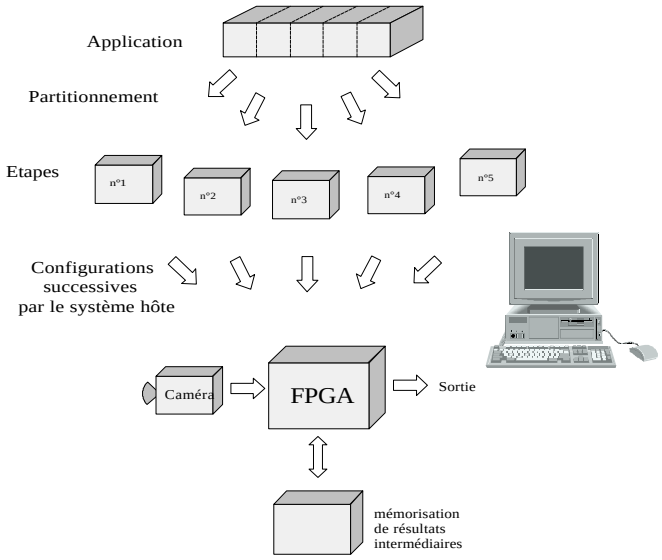

Fig. 2 : Principe de la reconfiguration dynamique globale.

La ***reconfigurabilité dynamique partielle*** se démarque des autres, par l'aspect partiel de la reconfiguration du circuit. Le principe, présenté par la figure 3, consiste à ne modifier qu'une partie du circuit ce qui permet de réduire le temps de configuration tout en donnant une souplesse accrue au système. On parle alors quelquefois de processeurs mous ou flexibles. En outre, la reconfiguration partielle peut intervenir pendant le fonctionnement du reste du circuit.

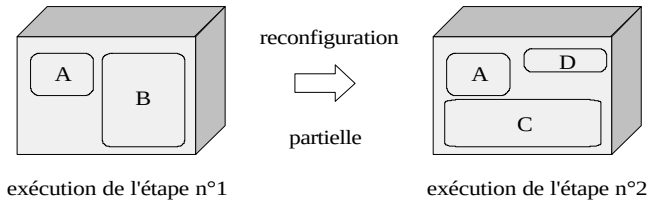

Fig. 3 : Principe de la reconfiguration dynamique partielle.

2.2 Systèmes existants

Les travaux de recherche sur les FPGAs reconfigurables ont conduit à deux grandes familles de systèmes :

- Les ***processeurs reconfigurables*** sont dotés d'un cœur de processeur et d'un jeu d'instructions très réduit mais évolutif. Le principe consiste à développer des instructions spécifiques à l'application qui seront optimisées pour obtenir les performances souhaitées. La reconfiguration est souvent statique et seuls quelques prototypes tentent de mettre en œuvre la reconfiguration dynamique [1] [2] [3].

- Les ***accélérateurs d'algorithmes*** sont générés autour d'une machine à microprocesseur d'usage général, d'une plate-forme reconfigurable à base de FPGAs et d'un système d'échange de données à très haut débit. Le principe consiste alors à délester le microprocesseur de tous les calculs très répétitifs ayant besoin d'être effectués à haute vitesse. Ce dernier ne traite alors que de faibles volumes de données. Là encore, la littérature ne présente quasiment que des systèmes exploités de manière statique. [4] [5] [6].

3.Traitement d'image et reconfigurabilité

Deux caractéristiques nous encouragent à étudier l'exploitation de la reconfigurabilité dynamique des FPGAs dans le domaine du TIBN.

La première est la quantité de travaux menés avec succès pour implanter des algorithmes de TIBN sur des architectures à base de FPGAs sous forme de traitements en flot de données en temps réel.

La seconde est la succession d'algorithmes qui caractérise les chaînes de TIBN. Cette constitution naturelle correspond, a priori très bien, à la phase de partitionnement de l'application qui est nécessaire à l'exploitation dynamique de la reconfigurabilité des FPGAs.

3.1 Contraintes sur l'architecture

Nous présentons ici, à la fois les contraintes et leurs origines, ainsi que nos choix pour apporter une solution aux problèmes posés. Nous décrivons ainsi l'architecture retenue.

L'enchainement de plusieurs phases de configuration et de traitement pour réaliser une chaîne de traitement d'image en 40 ms (cadence vidéo) nécessite la désynchronisation du flot vidéo numérisé et du traitement. Aussi, la sauvegarde d'image source est nécessaire. Ces mémoires d'images seront aussi nécessaires à la mémorisation de résultats intermédiaires, de taille et de nature évolutives, dus à l'enchainement des phases de traitement. Ainsi, nous dotons l'architecture de 2 plans mémoire d'image 512 x 512 x 16 bits pilotés indépendamment et qui peuvent être exploités sous la forme de 4 plans 512 x 512 x 8 bits ou encore 8 plans 512 x 512 x 4 bits.

Le TIBN en flot de données sur des fenêtres locales impose, en outre, l'accès simultané aux pixels de plusieurs lignes successives de l'image. Nous avons choisi des mémoires FIFOs (512 x 8 bits) pour répondre à ce besoin en sachant que d'autres solutions ont été proposées. Notamment, le projet ARDOISE qui débute au sein du GDR ISIS propose une organisation assez sophistiquée des données dans des mémoires à double port et à bus large pour accéder aux pixels de plusieurs lignes en un seul accès à la mémoire.

Enfin, une mémoire de travail sera associée à ces ressources en mémoires pour la sauvegarde de données nécessaires à deux étapes successives et qui seraient d'une nature autre qu' une image.

Les mémoires FIFOs, au nombre de 16, sont réparties sur 2 FPGAs XC3195 dédiés au traitement proprement dit de l'information. Elles sont pilotées par 2 FPGAs XC3164, dits de gestion, qui prennent aussi à leur charge les plans mémoire d'image.

Enfin, 2 FPGAs XC3090, dits de communication, facilitent les échanges de données entre les FPGas de traitement et les mémoires d'images. Le cas échéant, ils auront la possibilité d'effectuer quelques traitements simples.

La figure 4 présente l'architecture que nous proposons. Elle est conçue pour présenter un maximum de souplesse d'utilisation. Elle pourra alors satisfaire la caractéristique très

forte d'Adéquation entre l'Algorithme et l'Architecture qui s'impose au fur et à mesure de l'implantation des différentes étapes d'une chaîne de traitement.

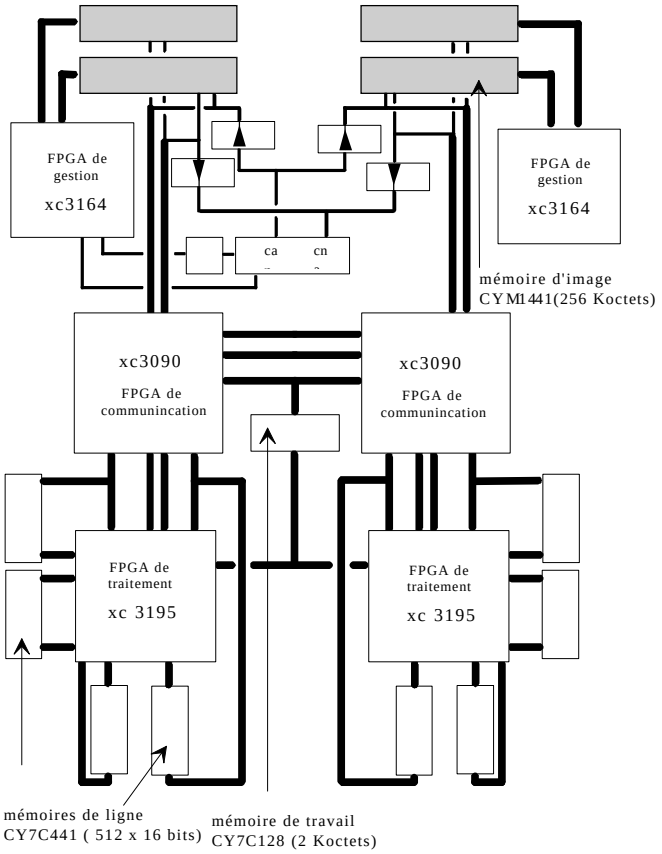

Fig. 4: Architecture proposée pour l'exploitation dynamique de la reconfigurabilité des FPGAs.

3.2 Réalisation

Nous avons ensuite chercher à valider cette architecture en implantant la chaîne de segmentation de D. Demigny [7].

Le filtres de Nagao et le calcul du gradient (norme et direction) ont été mis en œuvre physiquement sur une carte prototype pour former une première étape.

Les algorithmes de suppression des non maxima locaux, de double seuillage, d'amélioration des contours, d'affectation des extrémités et d'étiquetage des crêtes ont été décrits et simulés dans une seconde étape.

L'algorithme de résolution des équivalences entre contours forme une troisième étape.

Enfin, l'étiquetage des régions passe, comme pour la fermeture des contours, par 2 étapes d'étiquetage des régions et de résolution des équivalences entre régions.

Cette recherche d'implantation nous a permis de valider notre architecture pour sa souplesse et sa capacité à recevoir les différents algorithmes d'une chaîne de traitement d'image bas niveau.

C'est bien l'organisation des FPGAs et des mémoires que nous validons, puisque la série XC3000 de Xilinx utilisée, ne permet pas d'exploiter dynamiquement la reconfigurabilité, dans la mesure où les circuits sont trop lents en reconfiguration.

Dès le début de notre travail, l'idée était d'étudier la reconfiguration dynamique, de proposer une architecture et de la valider dans son principe, pour pouvoir la reprendre dès que des circuits supportant une reconfiguration très rapide, partielle et en cours de fonctionnement seraient disponibles. D'autres équipes de recherche au sein du GDR PRC ISIS ont travaillé à la proposition d'architecture. Sans être exhaustifs, nous citerons les travaux du LIEN de Nancy et du laboratoire ETIS de Cergy Pontoise. Ces travaux ont participé à l'animation des réunions de l'opération 7.3 du GDR ISIS et ont débouché sur une action incitative: le projet ARDOISE.

3. Evaluation des performances

Face aux différentes stratégies d'implantation d'une application dans le cadre de la reconfiguration dynamique des FPGAs, l'exploitation dynamique de la reconfigurabilité est–elle, dans tous les cas, à la fois performante et économique par rapport à une implantation complète de l'application sous forme statique ?

Pour répondre à cette question, nous utilisons un critère de mesure, la ***Densité Fonctionnelle*** (*DF*), qui prend en compte les coûts en ressources matérielles et en temps. Une définition rigoureuse de la *DF* nécessite de prendre en compte le coût matériel que représentent les ressources mémoires et le système de gestion de la reconfiguration dynamique. Après quelques simplifications, la *DF* s'exprime par la relation (1) où *Tc* est la durée totale des reconfigurations du système, *Te* la durée effective des exécutions des algorithmes et A le coût en CLBs de l'implantation de l'application. Nous pouvons alors mesurer le nombre d'images traitées par seconde et par CLB.

$$DF = \frac{1}{A(T_e + T_c)} \qquad (1)$$

Pour une application donnée, l'estimation de la *DF* dans chacun des cas statique et dynamique fournira DF_{stat} et DF_{dyn}. La DF_{dyn} est maximale pour $Tc = 0$ et s'exprime par la relation (2).

$$DF_{dyn\max} = \frac{1}{AT_e} \qquad (2)$$

La relation (3) exprime alors la *DF* normalisée, qui est définie en fonction du rapport r de *Tc* sur *Te*.

$$\frac{DF_{dyn}}{DF_{dyn\max}} = \frac{1}{1+r} \qquad (3)$$

Nous en déduisons que le rapport r est crucial pour exploiter au maximum le potentiel de la reconfiguration dynamique et qu'il devra donc rester le plus faible possible.
Il faut noter que réduire T_e , seul, fait croître DF_{dyn} , mais aussi DF_{dynmax} , ce qui rend T_c plus prépondérant encore, dans le coût temporel total. Ainsi, il est préférable de réduire T_c.

Les solutions pour réduire le rapport r sont peu nombreuses :

- réduire la valeur intrinsèque de T_c . C'est le travail du constructeur ;
- mettre en œuvre la reconfiguration partielle. Elle permettra, par exemple, de conserver une partie des modules communs à deux configurations successives, et par-là, de réduire le temps de configuration ;
- chercher à réduire l'effet de T_c par rapport à T_e. Cet objectif peut être atteint en s'assurant que pour une configuration donnée, un maximum de calculs seront effectués. T_e pourra alors augmenter pour une valeur de T_c qui restera constante ;
- enfin, l'aspect multi-FPGAs d'une architecture permettra de procéder à plusieurs téléchargements en parallèle au lieu de configurer les circuits de manière complètement séquentielle. La durée du téléchargement sera réduite au temps de téléchargement le plus grand parmi les différents circuits utilisés.

Finalement, pour toute application, il faudra estimer la DF_{stat} et la $DF_{dyn\ max}$. La reconfigurabilité dynamique sera d'autant plus intéressante, potentiellement, que la $DF_{dyn\ max}$ sera très supérieure à la DF_{stat}. La figure 5 illustre ce potentiel en représentant la densité fonctionnelle normalisée en fonction de r. L'estimation de la DF_{stat} permettra de déterminer la valeur limite de r , au delà de laquelle la mise en œuvre de la reconfigurabilité dynamique sera globalement pénalisante par rapport à une implantation en reconfiguration statique.

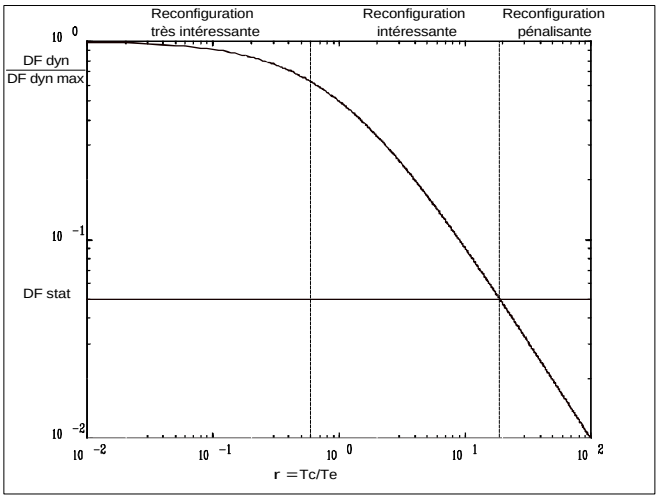

Fig. 5: Densité fonctionnelle normalisée en fonction de r

Si nous nous limitons au cadre du projet ARDOISE, la *DF* perd quelque peu de son intérêt dans la mesure où nous souhaitons, quoi qu'il arrive, travailler en temps réel. Ainsi, les implantations, qu'elles soient en reconfiguration statique ou dynamique, doivent avant tout répondre à cette contrainte de temps réel. Dès lors, la reconfiguration dynamique, qui présentera toujours une économie de ressources de calculs par rapport à une implantation statique, conduira systématiquement à une DF_{dyn} supérieure à la DF_{stat}.

En revanche, dans le domaine des accélérateurs d'algorithmes où la contrainte de temps réel n'est pas imposée, le critère de *DF* reprendra tout son sens et permettra d'évaluer l'opportunité de mettre en œuvre l'exploitation dynamique de la reconfigurabilité des FPGAs. Elle le fera en mettant en regard les surcoûts matériels et temporels engendrés par les différentes stratégies d'implantation.

4. Conclusions

Après l'étude des caractéristiques de l'exploitation de la reconfigurabilité des FPGAs dans le domaine du traitement d'image, nous avons proposé une architecture susceptible de recevoir les différents algorithmes d'une chaîne de traitement bas niveau. Nous avons alors validé cette architecture pour sa souplesse, en implantant les différents algorithmes de la chaîne de segmentation développée par l'équipe de D. Demigny du laboratoire ETIS [7].

Enfin, un critère de mesure, la Densité Fonctionnelle, est proposé pour comparer les différentes stratégies d'implantation d'une application et évaluer par là l'opportunité de mettre en œuvre l'une ou l'autre de ces stratégies.

Références

[1] M.J. Wirthlin, B. L. Hutchings, K.L. Gilson : *The nano processor : a low resource reconfigurable processor*. IEEE workshop on FPGAs for custom computing machines, Los Alamitos, California, 10 – 13 april 1994, pp 23 – 30, in Duncan A. Buell and Kenneth L. Pocek editors, IEEE computer society, 1994

[2] J. Davidson *FPGA implementation of a reconfigurable microprocessor* in proceedings of the IEEE 1993 custom integrated circuits conference, pp 3.2.1 – 3.2.4, 1993

[3] Mickael J. Wirthlin, Brad L. Hutchings: *A Dynamic Instruction Set Computer*, IEEE Workshop on FPGAs for Custom Computing Machines, Napa, California, 19 - 21 avril 1995

[4] J.G. Eldredge, B. L. Hutchings *RRANN : a hardware implementation of the backpropagation algorithm using reconfigurable FPGAs*, IEEE world conference on computational intelligence, Orlando, Florida, pp 77 – 80, 26 june – 2 july 1994,

[5] J. M. Arnold, D. A. Buell, E. G. Davis, *Splash 2* proceedings of the 4th annual acm symposium on parallel algorithms and architecture, pp 316 –324, june 1992

[6] P. M. Athanas, H. Silvreman, *Processor Reconfiguration throught Instruction Set Metamorphosis*, IEEE computer, march 1993

[7] J. F. Quesne, *Vision robotique : architecture data-flow pour le traitement des images en temps réel*, Thèse de doctorat de l'Université de Paris-Sud (Centre d'Orsay), 1992