Reconfiguration dynamique des FPGA pour une segmentation d'image adaptative et temps réel

Ryad Bourguiba, Lounis Kessal, Didier Demigny

ETIS, URA D2235,
ENSEA/UCP, 6 avenue du Ponceau
95014 Cergy Pontoise Cedex, France
demigny@ensea.fr

RÉSUMÉ

Les évolutions des systèmes de communications numériques, notamment dans les domaines du traitement des images, requièrent des algorithmes de plus en plus complexes nécessitant des puissances de calcul au delà des performances des processeurs actuels. Nous montrons que la reconfiguration dynamique des FPGA autorise sur une même architecture matérielle faible coût, l'exécution séquentielle des différents algorithmes constituant une chaine de segmentation et ceci, sous la contrainte temps réel. Nous discutons les concepts clés d'une architecture performante : synchronisation, couplage mémoires/FPGA, reconfiguration dynamique, modèle informatique.

ABSTRACT

Real time image processing required more and more complex algorithms which need power computation not available with current processors. We show that the fast reprogrammability of the FPGA allows the sequential execution of the different algorithms which compute an image segmentation on a low cost hardware architecture. We discuss the key concepts for the definition of an efficient architecture : synchronization, memory/FPGA link, reprogrammability, conceptual model.

1 Introduction

L'évolution des systèmes de communication numériques, notamment dans les domaines du traitement et de la transmission des images, requièrent des algorithmes de plus en plus complexes nécessitant des puissances de calcul au delà des performances des processeurs actuels. Deux grandes approches sont actuellement utilisées :

- la mise en parallèle de processeurs qui apportent la souplesse de programmation et la rapidité, mais au détriment du coût et de l'encombrement ;
- les circuits spécifiques (ASIC) qui apportent la rapidité de traitement et le faible coût, mais au détriment de la souplesse d'adaptation des algorithmes.

Aujourd'hui, l'émergence de la technologie FPGA laisse la place à une solution intermédiaire où on allie flexibilité de la programmation et puissance de calcul des architectures spécialisées. C'est un paradigme nouveau - et maintenant reconnu - à partir duquel le matériel devient malléable : *spongy computer*. Dans le cadre du GDR ISIS (opération 6.3), nous avons initié une réflexion sur la conception d'une structure mixte FPGA/DSP. L'ambition est de pouvoir enchaîner les exécutions d'un flot d'algorithmes de segmentation d'images sur une même structure physique (Fig.1). Si les processeurs classiques ou les DSP remplissent cette tâche simplement, c'est au détriment de la vitesse ou bien grâce à un taux de parallélisme conséquent (qq. dizaines de processeurs).

Pour un unique processeur et un unique algorithme, un ordre de grandeur est à gagner en rapidité pour assurer un débit

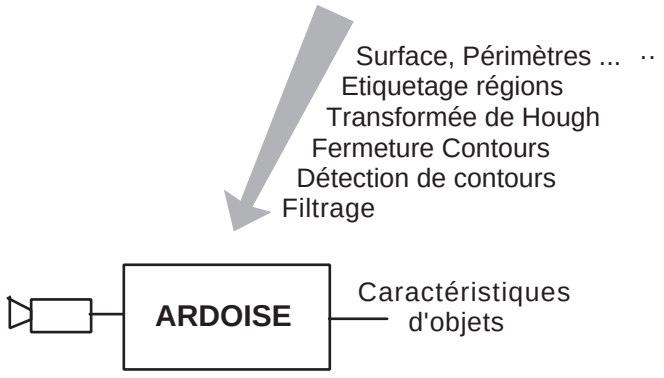

FIG. 1 — Reconfiguration dynamique

temps réel d'images (25 images par seconde). Les FPGA basés sur les technologies SRAM, par leur possibilité de reconfiguration, permettent la modification des traitements qu'ils exécutent tout en conservant les performances de vitesse ($\simeq 1/3$ perf. ASIC). Les complexités de traitement actuellement envisageables sont tout à fait adaptées aux algorithmes utilisés en segmentation. La reconfiguration a largement été employée sous un aspect statique (on choisit l'algorithme exécuté mais on ne le change pas en cours d'utilisation) parce que les durées de reconfiguration étaient prohibitives ($\simeq 100ms$). Les progrès technologiques et la volonté des fabricants d'attaquer le marché des coprocesseurs programmables permettent d'envisager des temps de reconfiguration inférieurs à la milliseconde. Ceci rend concrétisable notre réflexion initiale. La reconfiguration dynamique permet à un réseau de faible taille (2-

4 circuits) de réaliser une succession de traitements telle que filtrage et détection de contours, suppression des non maxima locaux, fermeture de contours, étiquetage de région à une cadence de 25 images 512×512 par seconde. D'autre part, une coopération intelligente entre le bas niveau et l'interprétation d'image est facilement exploitable avec notre modèle d'exécution. Des algorithmes de caractérisation du type d'image peuvent être chargés dans les FPGA et aider à la sélection des opérateurs de segmentation et à leur paramétrage le plus pertinent. Ceux ci étant exécutés par la suite sans surcoût matériel.

A partir de quelques remarques sur la nature et la complexité des algorithmes à implanter, sur le couplage mémoire/FPGA, sur le séquencement des configurations/exécutions, nous établissons quelques directives pour une architecture système efficace **: Architecture Reconfigurable Dynamiquement Orientée Image et Signal Embarquable (**ARDOISE**).** Ce projet implique plus d'une dizaine d'équipes de recherche. Nous décrivons ici notre contribution à l'ébauche de ce projet.

2 Variété des algorithmes

De façon à aider à la définition de l'architecture, nous avons défini un ensemble d'algorithmes suffisamment différents en terme de modèle d'exécution (pipeline, flot de données, SIMD, parcours de listes) et de type de mémorisation de façon à dégager une architecture la plus souple possible tout en restant réaliste. Les algorithmes choisis sont représentatifs de la segmentation et ont fait l'objet d'études architecturales ASIC ou FPGA dans notre équipe : (histogrammeur, filtre de Nagao[1], filtre de Deriche[2], fermeture de contours[3], étiquetage de régions[4], suppression des non maxima locaux, polygonalisation de contours, transformée de Hough.

2.1 Puissance de calcul, nombre de portes

Pour évaluer la puissance de calcul, nous donnons une évaluation en MIPS et nous utilisons aussi l'unité définie par Jean Vuillemin : le BOPS pour Bit Operation per Second. Un BOPS correspond approximativement à l'exécution d'une addition 1 bit en une seconde. Cette définition est plus proche de l'association des calculs aux différents blocs logiques qui composent un FPGA. La table 1 précise les puissances de calcul et la complexité en nombre de portes équivalentes nécessaires à l'exécution de quelques uns des algorithmes précédemment cités. Ces résultats sont basés sur le traitement d'images 512×512 à la fréquence pixel de 10Mhz.

Table 1

Opérateurs	MIPS	GBOPS	nb de portes
Nagao	600	5,2	10K
Convolueur 17×17	700	9,4	10K
Contours	120	2,3	2K
Fermeture	3600	6,2	8K
Etiquetage	100	3,1	10K
Deriche	320	8	12K

Pour évaluer la puissance de calcul, seuls interviennent les blocs logiques de la partie opérative, bien que la fréquence d'horloge soit la même pour les différents algorithmes, on ne peut établir de relation directe entre GBOPS et nombre de portes. L'importance du séquencement, des stockages de données, des gestions d'adresse influent sur cette relation. La fermeture de contours utilise un algorithme basé sur les automates cellulaires nécessitant 32 itérations par pixel codé sur 2 bits. On peut noter l'inadéquation des processeurs classiques à effectuer ce genre de traitement par un écart relatif plus important entre les puissances en MIPS et en BOPS.

Des évaluations récentes ont montré que des processeurs tels que le Pentium à technologie MMX étaient capables d'exécuter un algorithme de Deriche en une centaine de millisecondes. Il serait alors tentant de prédire la fin des architectures spécifiques dans quelques années. Il faut cependant noter que d'une part un ASIC et même un FPGA peuvent à l'heure actuelle intégrer et exécuter en temps réel plusieurs algorithmes constituant la chaine de segmentation ; que d'autre part, ces réalisations cablées ont été optimisées pour la surface une fois le critère temps réel atteint. Tous ces algorithmes peuvent être pipelinés d'avantage (à coût en surface nul sur des FPGA) afin d'augmenter la vitesse de traitement d'un facteur 2 à 4 avec les technologies actuelles.

Pour estimer les potentialités des FPGA nous prendrons deux exemples issus de familles de composants Xilinx.

Le circuit XC4025 contient 1024 blocs logiques ($\simeq 25K$ portes), à la fréquence relativement faible de 10Mhz il offre une puissance crête de 20 GBOPS. Cette famille souffre d'un temps de reconfiguration trop important $\approx 100ms$ qui limite fortement son utilisation pour notre projet.

Le circuit XC6264 contient 16K cellules. Bien que le constructeur annonce une complexité équivalente à 100K portes, celle-ci est certainement plus réduite du fait qu'une dizaine de cellules sont nécessaires à la réalisation d'un additionneur 1 bit. A 10Mhz, 16 GBOPS au maximum seront disponibles. Cette nouvelle famille conçue spécialement pour réaliser des coprocesseurs programmables possède des temps de reconfiguration inférieurs à la milliseconde.

On peut donc conclure qu'il sera possible d'exécuter simultanément plusieurs algorithmes sur un même circuit. le problème du partionnement d'un algorithme sur plusieurs circuits ne devrait pas se poser. L'utilisation d'un nombre très réduit de circuits (2 à 4) doit permettre de satisfaire la contrainte temps réel.

2.2 Fonctionnalités et débits mémoires

On doit distinguer les stockages et débits liés aux entrées/sorties d'images de ceux supplémentaires inhérents à l'organisation interne d'un traitement. Pour les E/S d'image, tous les algorithmes étant de type flots de donnée, le stockage entre algorithmes est facultatif sauf dans le cas où une reconfiguration intervient entre deux traitements. La table 2 précise, pour les stockages en sortie, la largeur des mots en bits, la taille en Koctets et le débit en Moctets/seconde pour des images

512×512. Pour des images plus grandes, taille et débit sont à augmenter proportionnellement au nombre de pixels.

Table 2

Opérateurs	largeur	taille	débit
Nagao	12	384	15
Convolueur 17×17	20	640	25
Contours	2	64	2,5
Fermeture	1	32	1,25
Etiquetage	8	256	10
Deriche	20	640	25

La table 3 est relative aux besoins de stockage internes aux traitements. Différents types de mémoires sont utilisés : fifos lignes (f), lignes (l) nécessitant un adressage particulier, à accès aléatoire non régulier (a), image (i). Dans un soucis de compacité, nous ne détaillons pas les besoins pour les différents modèles. Nous indiquons en T les types, en MA le nombre de générateurs d'adresses, en L la largeur totale en bits, en TA la taille en Koctets et en D le débit en Moctets/seconde.

Table 3

Opérateurs	T	MA	L	TA	D
Nagao	f	1	32	2	80
Convolueur 17×17	f	1	128	8	320
Contours	f	2	36	2,3	90
Fermeture	f	1	64	4	160
Etiquetage	f,a,i	3	44	395	100
Deriche	l,i	2	36	386	90

On peut noter la grande variabilité des besoins pour chacun des paramètres. La nécessité d'une mémoire d'image se retrouve aussi pour le stockage de l'espace des paramètres de la transformée de Hough. Il est rassurant que les débits les plus importants correspondent aux mémoires de plus faible taille. L'évolution des FPGA permettra l'utilisation de leurs ressources internes pour ces stockages. Une architecture universelle devra cependant devoir s'adapter aux disparités de largeur de mots et à la multiplicité des modes d'adressage simultanés.

3 Modèles de séquencement

- Les algorithmes sont des processus.

Cette réflexion est primordiale. Elle établit un parallèle avec la gestion des systèmes temps réel classiques. Le contrôle global d'enchainement des algorithmes est assuré par un OS temps réel qui gère les commutations d'algorithmes et les synchronisations. Le chargement d'une nouvelle configuration est équivalent au chargement du code d'un nouveau processus. Dans le cas d'une répartition de traitements sur différents circuits, l'OS gère aussi les communications entre les traitements. Même si un modèle simpliste peut être utilisé dans les premières phases du projet, cette vision système offre un cadre à la sophistication du contrôle. On peut penser à une hiérarchie de processus exécutés sur un même circuit et le remplacement d'un seul des processus par reconfiguration partielle du FPGA. Il découle de cette réflexion quelques choix plus précis sur les modèles de séquencement.

- Acquisition vidéo et traitements doivent pouvoir être effectués à des cadences différentes.

Le standard Vidéo pour une image 512×512 impose une fréquence trame de 25Hz. Sur les 40 ms, seuls 26,2ms correspondent à l'arrivée de pixels utiles. Sans désynchronisation, la fréquence d'horloge est alors de 10Mhz. Si on répartit le traitement des pixels sur les 40ms, la fréquence d'horloge peut être abaissée à 6,5Mhz. A l'inverse, avec une horloge de seulement 26Mhz, on peut enchainer en séquence 4 traitements en respectant la cadence temps réel de 40ms par trame. Il est ainsi possible d'optimiser la durée de chaque traitement en utilisant la vitesse d'horloge maximale compatible avec l'implantation de l'algorithme. On peut aussi minimiser l'augmentation des fréquences d'horloge face à l'augmentation des tailles d'image. Enfin l'architecture devient plus adaptatable aux traitements 1D : codage, compression etc ou aux séquencements moins réguliers (étiquetage).

- La synchronisation flot vidéo / traitements est réalisée au niveau trame.

Il est ainsi possible de sous échantillonner le flot d'entrée pour une durée globale des traitements supérieure à la période trame. Les mémoires d'image assurent le maintien des données entre traitements (au même titre que des registres dans un processeur).

- La reconfiguration n'est qu'une succession d'accès mémoire.

C'est une évidence quand on connait la structure des FPGA, mais il nous semble nécessaire de réduire l'importance accordée à cet aspect du point de vue de l'architecte. Les durées de configuration se réduisant maintenant à $\simeq 1ms$ pour la famille 6200 de Xilinx, il est inutile de développer des efforts architecturaux pour chercher à masquer ce temps bien inférieur à la durée d'un traitement.

4 Organisation architecturale

La figure 2 est un schéma de principe de l'organisation proposée. Par soucis de simplicité nous ne faisons pas apparaître la gestion du flot d'entrée et des horloges. Le nombre de FPGA (4) est un compromis complexité/coût/rapidité. Chaque circuit dispose d'une mémoire locale ML. Une mémoire d'image partagée MP entre FPGA permet la mémorisation inter-traitements (pendant les configurations pour un même FPGA ou pipeline entre FPGA). Le DSP a un double rôle. D'une part, l'ensemble des FPGA peut être vu comme un processeur superscalaire, le DSP (Texas 320C40) en assurant le contrôle et les reconfigurations. D'autre part, il contribue à l'exécution de certains traitements pour lequel il est plus adapté que les FPGA.

Bien que notre premier but soit la conception d'un unique module, l'architecture proposée permet la cascadabilité.

Notons que le problème de l'adéquation algorithme archi-

tecture est reporté au niveau de la programmation des circuits. Le choix flot, pipeline, séquentiel, SIMD, MIMD devient un style de programmation dans un langage de haut niveau tel que VHDL.

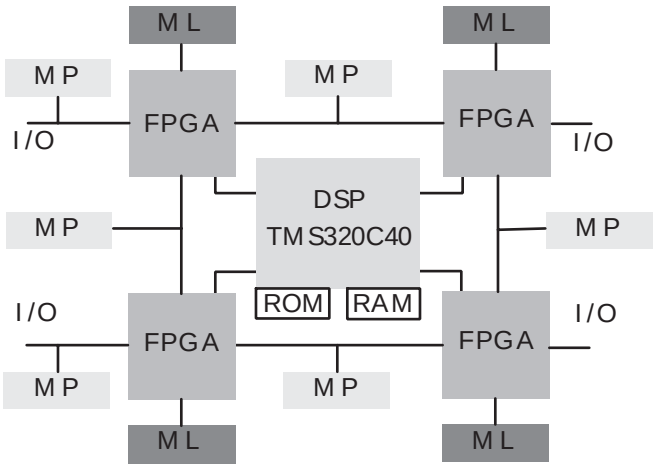

FIG. 2 — Architecture ARDOISE

5 Couplage mémoire/FPGA

C'est le point critique de la réussite de ce projet. Nous nous bornons ici à proposer quelques pistes qui seront précisées dans la version finale. L'objectif est qu'une structure mémoire conventionnelle puisse servir la variété des modes mémoires et assurer les débits nécessaires.

- Adaptation à la technologie FPGA

La faible vitesse de transfert est compensée par la disponibilité d'un grand nombre de connexions qui permet des bus mémoires larges.

- Variabilité, débit

L'accès simultané avec des modes d'adressage différents peut être remplacé à débit identique par un séquencement temporel des accès à une mémoire plus large. Par exemple, les accès réguliers (FIFO, images) sont effectués sur des chemins de données plus larges, ce qui libère du temps pour des accès aléatoires, puisqu'on lit en une fois les informations nécessaires à plusieurs cycles de traitement. On réduit ainsi le nombre de bus d'adresses différents. On a donc un comportement identique aux RAM vidéo, le registre à décalage étant ici interne aux FPGA, ce qui assure des fréquences d'entrées/sorties plus faibles.

- Exploitation des mémoires internes aux FPGA

On profite des capacités internes de mémorisation pour compenser les limitations du schéma régulier d'interfaçage des mémoires externes.

6 Outils

Les évolutions des outils de synthèse (optimisation logique, placement et routage) autorisent une programmation efficace des algorithmes en VHDL. Nos expériences ont montré qu'il était possible d'utiliser 100% des ressources d'un FPGA avec une efficacité comparable à une entrée du type schématique, sans intervention manuelle lors des phases de routage. Il faut moins de deux jours pour implanter un algorithme tel que le filtre de Nagao. La majeure partie du temps est passée en amont sur l'optimisation algorithmique. D'un point de vue système, il est nécessaire de développer un OS temps réel adapté qui sera exécuté sur le DSP pour assurer le séquencement des différents traitements. De façon plus prospective, on peut envisager un OS réparti sur les FPGA ; ceux-ci gérant eux même leur reconfiguration ... partielle.

7 Conclusion

Jusqu'à maintenant les solutions FPGA pour l'analyse d'image se sont contentées de mimer les solutions ASIC. La reconfiguration dynamique leur donne une spécificité qui rend enfin compatible dans une même structure la souplesse des machines programmables et la vitesse des machines cablées.

Références

[1] Didier Demigny, Jean Devars, Lounis Kessal, and Jean François Quesne. Implantation temps réel du filtre de lissage d'images de nagao. *Traitement du signal*, 10(4) :319–330, 1993.

[2] Didier Demigny, Lounis Kessal, Federico Garcia Lorca, and Jean Pierre Cocquerez. Conceptions nouvelles du détecteur de contours de deriche. In *Actes des Journées Adéquation Algorithmes Architectures En Traitement Du Signal et de L'image*, pages 45–52, Toulouse, janvier 1996. CNRS, GDR ISIS, CNES.

[3] Didier Demigny, Jean François Quesne, and Jean Devars. Boundary closing with asynchronous cellular automata. In *Proc. IEEE Conf. on Computer Architecture for Machine Perception*, number 1, pages 81–88, Paris, december 1991. CNRS, ETCA, AFCET, IEEE Circuit and Systems.

[4] Jean François Quesne, Didier Demigny, Jean Devars, and Jean Pierre Cocquerez. Architecture temps réel pour la fermeture des contours et l' étiquetage des régions. In *Actes du 8ème congrès Reconnaissance des Formes et Intelligence Artificielle*, pages 65–80, Villeurbanne, novembre 1991. AFCET.