Mise en œuvre du lisseur de Deriche sur l'architecture reconfigurable dynamiquement ARDOISE

N. ABEL, L. KESSAL, D. DEMIGNY

ETIS - ETIS UMR CNRS 8051 - ENSEA / UCP - 6, avenue du Ponceau – 95014 Cergy Pontoise Cedex

abel@ensea.fr

Résumé – ARDOISE, permet d'expérimenter les concepts résultant de la reconfiguration dynamique des FPGA. Les critères d'optimisation des traitements sur cette architecture diffèrent de ceux utilisés pour les architectures classiques. Nous proposons une méthode qui permet de réaliser cette optimisation et mettons en évidence les similitudes avec l'optimisation des nœuds de calcul sur les microprocesseurs. On illustre par ailleurs quelques spécificités qui font l'intérêt d'ARDOISE.

Abstract *– ARDOISE is designed for FPGA dynamical reconfiguration experimentation. For this architecture, designs performances differ of the one of classical implementations. We propose an optimisation method fitted for ARDOISE and make the parallelism with microprocessors' software optimisation. We finish this study illustrating some of the original points of our architecture.*

1. Introduction

La Reconfiguration Dynamique (RD) consiste à faire varier au cours du temps la fonction réalisée par un calculateur. Cette technique, bien qu'elle n'en porte pas le nom dans ce contexte, est à la base de tous les microprocesseurs. En effet, à chaque instruction correspond une fonction du calculateur, et cette fonction peut être reconfigurée dynamiquement à chaque coup d'horloge.

Le projet ARDOISE, initié lors de l'action incitative Architectures Reconfigurables Dynamiquement (ARD) du CNRS a conduit une dizaine de laboratoires français à imaginer une architecture matérielle permettant d'expérimenter les nouveaux concepts résultant de la RD des FPGA.

Depuis sa finalisation, cette architecture a permis de valider le principe même de la RD qui, dans ce cadre, consiste à reconfigurer plusieurs fois un FPGA pour réaliser un calcul. Chacune des configurations prenant en charge une partie de ce calcul. La reconfiguration partielle des FPGA a, elle aussi, été expérimentée. Cette dernière permet à une partie du FPGA de réaliser des successions de traitements distincts pendant que le reste du composant exécute une tâche de fond.

Un des aspects qui reste à mettre en œuvre est l'utilisation d'ARDOISE en tant que co-processeur pouvant exécuter des macro-instructions comme, par exemple, le lissage d'une image, le calcul du gradient, la fermeture de contours, l'étiquetage des régions...

Cette approche basée sur la construction d'une bibliothèque de traitements permettrait à terme d'allier la puissance des architectures à base de FPGA (les traitements pouvant être fortement parallélisés) à la souplesse du logiciel (le choix du traitement pouvant être conditionné par la nature des données à traiter.)

On commence par décrire les traitements de base d'une chaîne d'extraction de contours, et plus particulièrement dans la suite, le lisseur de Deriche. L'expertise du laboratoire ETIS sur ce traitement permet d'expérimenter rapidement différentes implantations sur ARDOISE, et de dégager des critères de performance dans le contexte de la RD (1). On constate que l'utilisation des mémoires disponibles conditionne de manière très sensible ces performances (2). On propose ensuite une méthode permettant d'optimiser les stratégies d'accès aux mémoires (3) pour terminer en illustrant quelques point forts d'ARDOISE (4). Nous concluons sur les perspectives (5).

2. Adéquation entre ARDOISE et le filtre de Deriche

Une description d'ARDOISE permet de définir les bases de notre étude [1][2]

Son principe de fonctionnement permet de définir des critères de performance simples (FIG. 1), on démontre que dans le cadre de la RD, la principale caractéristique est le temps nécessaire à la configuration, et au calcul d'un algorithme.

Son architecture permet ensuite d'envisager un nombre restreint de mises en œuvres du lissage de Deriche. On peut les lister exhaustivement selon le nombre de reconfigurations nécessaires.

Les figures FIG. 1 et FIG. 2 illustrent un mode de fonctionnement d'ARDOISE. Les carrés grisés représentent des FPGA, et les autres des mémoires.

Sur cet exemple de fonctionnement le FPGA F1 procède au rangement de nouvelles opérandes et à la transmission de résultats entre la mémoire MA et le monde extérieur (par exemple une carte vidéo).

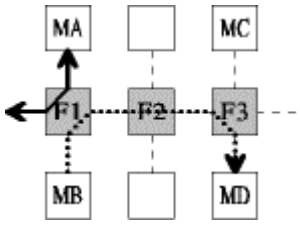

FIG. 1 : Calcul de T1 sur les données originales

La figure FIG. 1 illustre un traitement T1 implanté dans le FPGA F2 qui consomme ses opérandes dans MB, et stocke ses résultats dans MD.

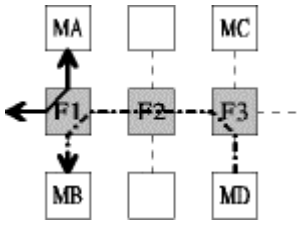

FIG. 2 : Calcul de T2 sur les résultats de T1

Sur la figure FIG. 2, F2 consomme les résultats précédemment produits dans MD pour stocker les résultats d'un nouveau calcul T2 dans MB.

Dans l'hypothèse où le traitement complet était la succession de ces deux traitements de base, on est en mesure, dès la fin du transfert de données entre MA et le monde extérieur, d'inverser les rôles de MA et MC avec ceux de MB et MD, et d'entreprendre le traitement sur les opérandes précédemment stockées dans MA.

Bien que cela ne se justifie pas sur cet exemple, le critère de performance choisi consiste à réaliser les deux traitements en un temps minimum. Pour ce faire, il faut minimiser le temps nécessaire aux reconfigurations et au calcul de T1 et T2. De cette manière, une fois ces traitements terminés, le temps restant peut être utilisé pour faire d'autres traitements.

Le filtre de Deriche est un traitement particulièrement bien adapté à sa mise en œuvre sur une architecture reconfigurable dynamiquement. En effet, on s'intéresse à l'implantation proposée par Federico Garcia-Lorca (FIG. 3). Cette implantation est une succession de traitements T et de stockage en mémoire. Chaque stockage en mémoire permet d'ordonner les données pour qu'elles soient traitées par le module suivant, et représente une opportunité de découper le traitement en plusieurs configurations. On distinguera les implantations selon qu'elles utilisent :

- une configuration (version originale).
- deux configurations, ie reconfiguration après le traitement horizontal (Hx).
- quatre configurations, ie reconfigurations après les traitements horizontal causal (Hc), horizontal anti-causal (Ha), vertical causal (Vc).

Le transfert du filtre T est donné par :

$$T(z) = (1 - \gamma)^2/(1 - \gamma z^{-1})^2 \qquad (1)$$

Un autre argument en faveur de l'utilisation de la reconfiguration dynamique pour réaliser le lisseur de Deriche tient au fait que le traitement effectué est toujours le même. Par conséquent, une grande partie du FPGA reste identique au cours des configurations. Le recours à la reconfiguration partielle permet de minimiser les temps de reconfiguration des FPGA : dans le cas présent, seuls les modules d'accès aux mémoires évoluent au cours du temps et nécessitent d'être reconfigurés.

3. Utilisation des mémoires

Lors de l'optimisation menée sur l'algorithme de Deriche, l'équipe ETIS avait abouti à une structure composée d'un FPGA (XILINX 4025), de deux mémoires lignes (Hc → Ha et Vc → Va : 512 mot de 16 bits) et d'une mémoire image (Ha → Vc : 512*512 mots de 16 bits). Cette étude était menée en vue de minimiser la structure matérielle nécessaire à ce traitement [3][4]. Or, dans la mesure où les contraintes ne sont plus les mêmes, il est légitime de revenir sur les choix précédemment faits.

Par exemple, dans la mesure ou ARDOISE ne dispose que de mémoires images, il n'est pas forcément avantageux d'avoir recours à des mémoires lignes (Implantation 4).

Sur la figure FIG. 3, les mémoires sont utilisées pour réordonner les données entre les différents traitements. Pour l'implantation 1, ce sont des mémoires intermédiaires : les données qu'elles contiennent n'ont d'importance que durant la phase de calcul (résultats intermédiaires). L'architecture flot de données proposée par Federico Garcia-Lorca consomme ses opérandes et fournit ses résultats vers la carte vidéo. Dans le cas d'une implantation sur ARDOISE, les opérandes et les résultats sont stockés dans les mémoires MA, MB, MC, MD (MGTI). Les mémoires blanches (MTGV) sont utilisées pour les stockages intermédiaires (FIG. 1, FIG. 2).

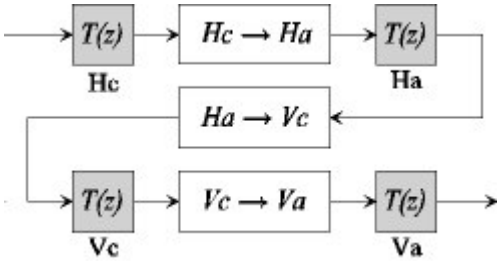

FIG. 3 : Implantation du lisseur de Deriche

La mise en ordre des données dans les mémoires lignes (Hc → Ha) implique une latence d'une ligne : cette mémoire est successivement lue dans le sens des adresses croissantes puis décroissantes. Ainsi, les opérandes de Ha sont toujours lues dans le sens inverse des résultats stockés par Hc (Causal → Anti-causal) pour la ligne précédente.

De la même manière, pour la mémoire intermédiaire (Ha → Vc), la latence introduite est d'une image. Dans ce cas l'adressage est plus complexe [3] Ainsi, pendant le traitement de l'image n, l'image stockée dans la mémoire (Ha → Vc) ne peut être effacée. Ceci peut devenir gênant si un traitement

ultérieur (ex : amincissement de contours, étiquetage de régions …) a besoin des emplacements mémoires utilisés.

Cette gêne est éliminée dès lors qu'on recourt à la reconfiguration dynamique. Pour les implantations 2 et 4, les résultats de Ha sont stockés dans les M_{GTI} et sont directement utilisés par la configuration suivante.

On remarque, par la même occasion que l'utilisation des mémoires diffère suivant les implantations. En particulier, les bandes passantes (en nombre de données de 16 bits lues ou écrites par cycle de traitement) nécessaires sur les M_{GTI} (M_{GTIE} pour l'entrée et M_{GTIS} pour la sortie) et les M_{TGV} (pour les résultats intermédiaires) sont données pour les trois implantations en fonction du taux de parallélisme P (TAB. 1). Le parallélisme est P lorsque P lignes (resp. P colonnes) sont traitées en parallèle pour les traitements horizontaux (resp. verticaux).

TAB. 1 : nombre d'accès par cycle de traitement

	M_{GTIE}	M_{TGV}	M_{GTIS}
Implantation 1	Nt_{GTI}=P	Nt_{TGV}=6*P	P
Implantation 2	P	2*P	P
Implantation 4	P	0	P

Ces résultats sont à confronter aux données accessibles par cycle de chaque mémoire (TAB. 2)

TAB. 2 : nombre d'accès par cycle d'adressage

M_{GTIE}	M_{TGV}	M_{GTIS}
Nm_{GTI}=2	Nm_{GTI}=4	2

4. Recherche du parallélisme optimal

Dans ce chapitre, nous proposons une méthode d'ajustement du parallélisme optimal au vu du critère précédemment donné. Le but est de réduire le temps de traitement. Deux dernières données sont nécessaires à cette étude (TAB. 3). Le temps d'accès aux mémoires est une donnée technologique, alors que la fréquence de traitement est propre à l'algorithme choisi. Dans notre cas, le caractère récursif du lisseur de Deriche empêche de diminuer le temps de traitement par recours à la technique du pipeline.

TAB. 3 : temps limites

Accès aux mémoires	Tm_{min}=20 ns	50 MHz
Cycle de traitement	Tt_{min}=37 ns	27 MHz

On se donne un dernier paramètre permettant d'ajuster les performances du traitement : le quotient (Qmt) de la fréquence d'accès aux données (Fm=1/Tm), et de la fréquence de traitement (Ft=1/Tt). Suivant ce quotient, la fréquence de traitement Tt va être déterminée soit :

- par les accès mémoire
  - $Qmt*Tm_{min}=Tt>Tt_{min}$
- soit par la fréquence maximale de traitement
  - $Qmt*Tm_{min}<Tt=Tt_{min}$

Par exemple, dans le cas de l'implantation 2, la mise en place d'un parallélisme d'ordre 2 est envisageable avec Qmt=1 ($Nt_{GTI}=Nm_{GTI}=2$ et $Nt_{TGV}=Nm_{TGV}=4$). Sous cette hypothèse, c'est la fréquence de traitement qui borne la bande passante vers les mémoires ($Tt=Tt_{min}$).

Pour un parallélisme d'ordre 4, il faut deux cycles (Qmt=2) pour accéder aux données ($Nt_{GTI}=2*Nm_{GTI}=4$ et $Nt_{TGV}=2*Nm_{TGV}=8$) nécessaires à un cycle de traitement. Dans ce cas, la période minimale de traitement ($Tt=2*Tm_{min}$) est imposée par l'adressage des mémoires.

Le tableau ci-dessous (TAB. 4) indique les résultats obtenus pour les différentes implémentations. Pour chaque valeur de Qmt, on détermine le taux de parallélisme maximum, la période de traitement minimale puis le temps nécessaire au lissage d'une image 512*512.

Pour ce dernier calcul, on émet l'hypothèse que le temps de reconfiguration d'ARDOISE est d'une milliseconde.

TAB. 4 : performances des différentes implémentations

Implémentation 1			
Qmt	Pmax	Tmin (ns)	T (ms)
1	0	37	
2	1	40	11,49
3	2	60	8,86
4	2	80	11,49
Implémentation 2			
Qmt	Pmax	Tmin (ns)	T (ms)
1	2	37	11,70
2	4	40	7,24
3	6	60	7,24
4	8	80	7,24
Implémentation 4			
Qmt	Pmax	Tmin (ns)	T (ms)
1	2	37	23,40
2	4	40	14,49
3	6	60	14,49
4	8	80	14,49

Il existe un temps minimum pour faire le calcul. Cette optimisation du critère qu'on s'était imposée est obtenue pour plusieurs taux de parallélisme. Cependant, l'augmentation du taux de parallélisme est accompagnée d'une augmentation de Qmt. Ceci complique la synchronisation des accès et des traitements et on choisit donc le parallélisme minimum qui aboutit à l'optimisation de notre critère.

Si pour des raisons d'économie de puissance, on était amené à chercher le taux de parallélisme maximum, c'est la surface utilisée par le module de traitement qui déterminerait P.

5. Avantages liés à l'utilisation d'ARDOISE

L'étude précédente permet de dégager plusieurs des points forts d'ARDOISE.

Pour commencer, le calcul du filtre de Deriche utilise moins d'un quart du temps nécessaire à l'acquisition d'une image (40 ms). Il est donc possible de reconfigurer ARDOISE afin d'utiliser le temps restant pour effectuer d'autres calculs. Le calcul de la norme du gradient, mis au point pour une démonstration de RD dans le cadre du filtrage de Sobel, peut naturellement suivre le lissage de Deriche.

Le lissage de Sobel étant à la fois plus rapide et moins performant, on peut décider en temps réel, suivant les conditions de bruit notamment, de se contenter du lisseur de Sobel en vue de mettre à profit le temps dégagé pour d'autres calculs ; ceci tout en étant capable de faire un nouveau compromis si les conditions l'exigent. En effet, le lissage de Sobel [5] peut être réalisé en moins de 5 ms.

Enfin, l'interface de configuration du FPGA permet de paramétrer le traitement sans avoir recours à une interface supplémentaire. A chaque valeur du coefficient γ peut correspondre une configuration. On peut ainsi se passer d'une partie de la logique (multiplexeurs réalisant les multiplications par γ). La diminution des temps de propagation (dûe à l'absence de multiplexeurs) est accentuée par une réduction des ressources de routage nécessaires à l'implémentation.

On passe ainsi d'une Fréquence maximale de traitement avec multiplexeurs de 17 MHz à une fréquence de 27 MHz sans multiplexeur.

6. Conclusion

L'optimisation du filtre de Deriche pour son implantation sur ARDOISE conduit à des choix nouveaux. On cherche dans ces conditions à utiliser le plus efficacement possible le matériel disponible, plutôt que d'économiser de la surface. On peut faire l'analogie avec l'optimisation de nœuds de calculs sur les processeurs parallèles : là aussi, l'idée est d'utiliser au mieux les différentes unités. Les performances sont alors limitées, soit par le nombre d'unités pouvant travailler parallèlement, soit par la bande passante d'acheminement des données vers ces unités.

D'une part, le grain fin des FPGA permet de réaliser des structures sur mesure pour les nœuds de calcul, et, d'autre part, le nombre de mémoires présentes sur ARDOISE, et leur grande liberté d'utilisation autorisent la mise en place des flux adaptés au traitement visé. En raison de cette grande liberté, l'optimisation des traitements est généralement longue.

L'autre analogie avec les microprocesseurs est de rendre la structure matérielle dépendante des données qu'elle traite. Les différentes configurations pouvant être vues comme des macro-instructions (équivalent d'un nœud de traitement) traitant un nombre important de données. Là encore, le temps nécessaire au développement de ces instructions empêche de réagir rapidement à un changement de contexte (à moins d'avoir prévu ce changement *ie* création d'une bibliothèque de traitements utiles).

Dans les deux cas, on se rend compte que pour se rapprocher encore plus des microprocesseurs, il faut être capable de renoncer provisoirement aux optimisations, et de pouvoir générer des nœuds rapidement. Pour faciliter cette première étape, on peut commencer par recenser les différents types de mémoires, et étudier leurs caractéristiques. On peut ensuite adopter la même démarche pour les modes d'adressage, et étudier leur implémentation sur les différents types de mémoires. L'idée étant, de faire cohabiter des générateurs d'adresse différents (généralisation des générateurs d'adresse des microprocesseurs) sur une ou plusieurs mémoires physiques.

Ces outils deviendraient une aide précieuse pour l'architecte utilisant un FPGA capable d'utiliser dans un premier temps des modes d'adressage standard sans avoir à les mettre en place.

Références

[1] R. Bourguiba. *Conception d'une architecture matérielle reconfigurable dynamiquement dédiée au traitement d'images en temps réel*. Thèse de doctorat de l'université de Cergy Pontoise. 7 juillet 2000.

[2] D. Demigny, L. Kessal, R. Bourguiba e N. Boudouani. *How to use high speed reconfigurable fpga for real time image processing ?* In Proc. IEEE Conf. on Computer Architecture for Machine Perception. Padova, september 2000. IEEE Circuit and Systems (pages 240-246).

[3] D. Demigny, F. Garcia Lorca et L. Kessal. *De l'architecture à l'algorithme. Un exemple: le détecteur de contours de Deriche*. Traitement du signal. volume 14 - n° 6. Spécial 1997 (pages 615 à 623)

[4] F. Garcia Lorca. *Filtres récursifs temps-réel pour la détection de contours : optimisations algorithmiques et architecturales*. Thèse de doctorat de l'université d'Orsay novembre 1996.

[5] N. Abel, D. Demigny, L. Kessal, N. Boudouani – *Mise en œuvre de la reconfiguration partielle sur l'architecture reconfigurable ARDOISE*. Conf. JFAAA'2002. E. Martin et M. Abid (eds.) (pages 45 à 48)