



Circuits Reconfigurables et Traitement Bas Niveau d'Images en Temps Réel

Marcelo ALVES DE BARROS et Mohamed AKIL

Groupe ESIEE - Laboratoire IAAI - BP 99, Cité Descartes, 2, Bd. Blaise Pascal
93162 - NOISY LE GRAND CEDEX - FRANCE - email: akilm @ apo.esiee.fr

RÉSUMÉ

Dans cet article, on présente l'étude d'un modèle d'architecture adapté pour le traitement bas-niveau des images. Pour ce modèle basé sur un pipeline d'opérateurs de base dédiés, nous étudions les problèmes concernant l'évaluation et la validation de l'implémentation sur des circuits reconfigurables. Nous présentons une méthode d'évaluation de coût en surface et performances temporelles d'un tel modèle d'architectures, ainsi qu'une étude de faisabilité d'une chaîne de pré-traitement.

ABSTRACT

This work presents a pipeline based model of architectures adapted to implement low level image operators. We discuss the guidelines to evaluate and validate the architectures implemented using this model and SRAM based FPGA. A method is described to compute area costs and time performances of the architectures and the exemple of a feasibility study is presented for a image pre-processing chain.

1. Introduction

Généralement, une implémentation matérielle d'algorithmes pour le traitement bas-niveau des images tente de respecter deux contraintes : une exécution de l'application en temps réel et une certaine flexibilité.

La contrainte de flexibilité, traduite par un certain degré de reconfigurabilité de l'architecture, impose un coût matériel additionnel important [Ruetz 86], [Fil 91] : unités arithmétiques et logiques programmables, systèmes d'interconnexions reconfigurables entre opérateurs et inter-opérateurs, ect...

De même le processus de validation engendre des coûts importants : en temps de conception et de fabrication [Wes 85].

La technologie reconfigurable, marquée par l'apparition de circuits FPGAs ("Field Programmable Gate Array") avec des performances sans cesse croissante en vitesse et en densité d'intégration, devient une option intéressante pour la réalisation de circuits dédiés temps réel, flexible et à faible coût pour le traitement bas niveau des images.

Dans ce cadre, nous proposons l'implémentation matérielle d'opérateurs pour le traitement bas niveau des images. Cette implémentation repose sur un modèle architecturale pipeliné, avec un mode d'opération synchrone en flot de données. Les exigences de performances sont déterminées par le débit. Nous justifions le choix de ce modèle par les aspects suivants :

1. les opérateurs opérant au niveau pixel, peuvent être successivement décomposés en sous-opérateurs jusqu'à des opérateurs arithmétiques ou logiques élémentaires au niveau mot, par l'insertion de couches additionnelles de pipeline. Le contrôle du degré de pipeline entre et inter opérateurs, permet le découpage des chemins critiques et l'adaptation de l'architecture aux débits exigés.

2. le moyen d'accès aux données de l'image le plus utilisé aussi bien en entrée qu'en sortie du traitement bas niveau est le balayage. Le modèle proposé favorise une uniformité de ces chemins de données.

3. le contrôle dans un tel type d'architecture est simple et peut être facilement généré à partir des signaux de synchronisme des sources d'images fournis par l'application.

L'approche proposée est représentée dans la Figure 1.1, et consiste en une cascade d'opérateurs de base, placés dans un réseau linéaire de circuits FPGA à technologie SRAM [Alves 92].

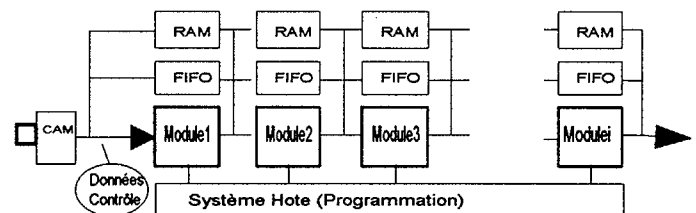


Figure 1.1

Nous proposons aussi une méthodologie d'évaluation des performances d'une telle implémentation. Cette approche permet donc une étude de faisabilité des algorithmes envisagés.

Dans la section 2, nous présentons succinctement les caractéristiques le technologie FPGA et en particulier celle programmable par SRAM de Xilinx. La section 3 présente une méthode d'évaluation des performances pour des opérateurs en traitement bas-niveau des images. La section 4 présente les résultats d'implémentation d'un ensemble d'opérateurs élémentaires et d'une chaîne de pré-traitement.

2. Circuits reconfigurables XILINX

Les circuits Xilinx appartiennent à la famille des circuits reconfigurables par SRAM. Le circuit XC4010 utilisé dans ce travail, comprend un réseau bidimensionnel de 400 Blocs Logiques élémentaires reconfigurables appelés CLB. Le CLB contient deux registres et réalise des fonctions combinatoires de 9 entrées et 2 sorties à travers la combinaison de 3 générateurs de fonctions : Look-Up-Table : G, F et H. La famille 4000, permet l'intégration de mémoire, en utilisant les bits de la mémoire de configuration du CLB, plus spécifiquement les bits



de la Look-Up-Table. Ainsi un CLB se configure en bloc mémoire de 32 x1bits ou 2x 16x1bits.

3. Evaluation des performances de l'implémentation d'opérateurs sur FPGA-Xilinx.

La méthode d'évaluation décrite ci-dessous, adaptée aux caractéristiques technologiques des FPGA-Xilinx permet de démontrer la faisabilité d'une classe importante d'algorithmes pour le traitement bas niveau des images. A partir du coût en surface d'un opérateur élémentaire, elle présente le coût de l'architecture algorithmique. De même qu'à partir de l'estimation de la performance temporelle d'un CLB et des interconnexions, on détermine le temps de réponse de l'ensemble de la structure de l'application.

3.1 Coût en surface d'un traitement élémentaire

Le modèle d'estimation de surface normalement utilisé comme référence [Wes 85] pour l'implémentation matérielle d'une architecture dédiée, sur une technologie VLSI, est donné par:

$$S_e = Nt/Ds$$

S_e est l'unité de surface du composant architectural élémentaire, normalement le FA (Full Adder), avec:

* Nt est le nombre des composants physiques élémentaires pour la technologie utilisée nécessaires pour la réalisation du FA, c'est à dire le nombre de transistors;

* Ds est la densité moyenne d'intégration permise par la technologie. Ce dernier paramètre est fonction des caractéristiques technologiques (largeur de canal), style de conception (standard cell, full custom), et de la régularité de la fonction élémentaire (FA).

Cependant, ce modèle s'avère inadapté à une technologie du type LCA de Xilinx. Ainsi, nous considérons un modèle moins générale avec une granularité plus large que le FA. Il s'agit d'utiliser comme composant architectural élémentaire un ensemble d'opérateurs élémentaires représentatifs de notre domaine d'applications. Chaque opérateur a un coût en surface élémentaire, donné en fonction du nombre de CLBs.

3.2 Coût en surface de l'architecture algorithmique

Soit E_{cc} , E_{mm} et E_{int} des quantités qui représentent respectivement "l'énergie" de calcul, de mémorisation et d'interconnexions de l'architecture. Le terme énergie est employé pour représenter les différents types d'opérateurs arithmétiques et/ou logiques de l'architecture. Soit A_{cc} , A_{mc} et A_{int} des quantités qui représentent respectivement le coût matériel (en surface) des composants élémentaires de calcul, de mémorisation et d'interconnexions. L'énergie de calcul est ici traduite par les énergies des composants de calcul qui caractérisent l'architecture tels que les comparateurs (E_{comp}), les additionneurs et soustracteurs (E_{add}), multiplexeurs (E_{mux}), etc. Le coût de ses éléments est exprimé en nombre de CLBs. Le coût matériel A_{arch} de l'architecture d'un algorithme donné peut être modélisé par l'équation :

$$A_{arch} = \left(\sum_i E_{cc_i} \cdot A_{cc_i} \right) + E_{mm} \cdot A_{mm} + E_{int} \cdot A_{int} \quad \text{Eq. 3.1}$$

où E_{cc_i} : énergie d'un élément de calcul i de l'architecture et A_{cc_i} : surface de l'élément de calcul i de l'architecture. L'équation adaptée à la technologie Xilinx est:

$$A_{x_{arch}} = \left(\sum_i E_{cc_i} \cdot A_{cc_i} \right) + E_{mm} \cdot A_{mm} + E_{mr} \cdot A_{mr} + E_{int} \cdot A_{int} \quad \text{Eq.3.2}$$

où E_{mr} et A_{mr} représentent respectivement l'énergie et la surface d'un registre.

3.3 Estimation de Vitesse

Si nous considérons la structure de la Figure 3.1 qui correspond à un opérateur de base ou un étage de pipeline du modèle d'architecture considéré, le temps de réponse d'une telle structure est donnée par:

$$T_c = N_{CCLB}(T_{CLB} + L \cdot T_{INT}) + T_{CKS} + T_{CKSK} \quad \text{Eq.3.3}$$

où:

N_{CCLB} : nombre de couches de CLBs; T_{CLB} : temps de propagation d'un CLB; L : longueur d'interconnexion dans le circuit pour chaque couche de CLB; T_{INT} : temps de propagation par unité de longueur d'interconnexion; T_{CKS} : temps de propagation horloge-sortie des registres; T_{CKSK} : temps de dispersion des horloges dans le circuit.

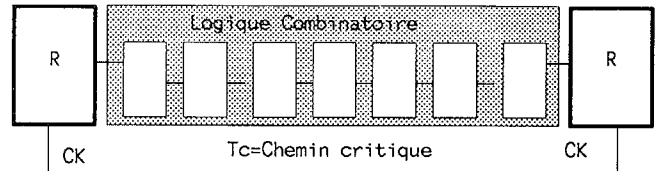


Figure 3.1

La longueur L d'une interconnexion est définie par le nombre de Points d'Interconnexion Programmables (PIPs) N_{PIP} et de Matrices d'Interconnexions N_{MI} traversés. A chaque élément d'interconnexion traversé, s'ajoute un délai élémentaire de propagation T_{INT} . Les quantités N_{PIP} et N_{MI} ne sont connues qu'après le routage. Cependant, ces deux paramètres peuvent être optimisés par un routage adapté, lequel se base sur une utilisation d'un maximum de 70% des CLBs. D'après nos expériences, une valeur moyenne de temps d'interconnexion ($N_{PIP} + N_{MI}$) entre couches de CLBs dans les opérateurs de base implémentés est 5 ns.

Pour les circuits de la famille XC4000-5 nous avons $T_{CKS} = 3$ ns. Si on considère le pire cas on a $T_{CLB} = 6$ ns. Si les lignes spéciales de contrôle ("globalnets") sont utilisées pour la distribution des horloges, le temps de dispersion est négligeable ($T_{CKSK} = 0$) [Xil 92]. Nous avons donc :

$$T_c = N_{CCLB}(6+5)+3+0=11N_{CCLB}+3 \quad \text{Eq. 3.4}$$

Pour estimer la profondeur maximale permise pour un opérateur donné, on a à partir de l'équation Eq. 3.3 :

$$N_{CCLB} = (T_c - 3) / 11 \quad \text{Eq.3.5}$$

Pour un traitement bas niveau d'images à la fréquence vidéo (10 MHz), nous avons:

$$N_{CCLB} = (100 - 3) / 11 = 8 \text{ couches} \quad \text{Eq. 3.6}$$

Au-delà de cette limite des optimisations architecturales telles que l'insertion des étages additionnels de pipeline et/ou la

parallelisation, sont nécessaires pour respecter le débit et/ou la fréquence exigée.

La latence L_{atp} de l'architecture est donnée par le nombre P d'étages de pipeline utilisés.

Dans le modèle pipéliné étudié, la fréquence maximale d'opération F_{max} , est déterminée par le temps de réponse de l'étage le plus lent du pipeline.

$$F_{max} = 1 / (\max[Tc_i]) \quad \text{Eq.3.7}$$

où Tc_i est le temps de cycle du i -ème étage de pipeline.

4. Implémentation des opérateurs élémentaires et d'une chaîne de pré-traitement

Les opérateurs que nous utilisons comme modèles de référence pour l'évaluation d'opérateurs de base à travers l'équation 3.2, sont donnés ci-dessous :

Opérateur élémentaire	Surface (CLBs)	Vitesse
ADD8 (additionneur 8 bits)	5	3T _{CLB}
SUB8 (soustracteur 8 bits)	5	3T _{CLB}
COMP8 (comparateur magnitude 8 bits)	5	3T _{CLB}
ID8 (comparateur d'identité 8 bits)	2	2T _{CLB}
REGV (registre V bits)	V/2	3 ns
REG8 (registre 8 bits)	4	3 ns
UCE8 (Unité compare/échange 8 bits)	13	4T _{CLB}
MUX8 (multiplexeur 8 bits)	4	1T _{CLB}

Tableau 4.1 : Famille XC4010-5 (TCLB = 6ns)

La chaîne de pré-traitement étudiée est montrée dans la Figure 4.1. La fonction de la chaîne est de générer des contours précis pour l'extraction des paramètres des objets. Ces paramètres sont comparés par une mise en correspondance, avec une bibliothèque de modèles préexistants, pour identifier l'objet courant. L'implémentation envisagée sur FPGA est représentée par la partie hachurée de la figure 4.1.

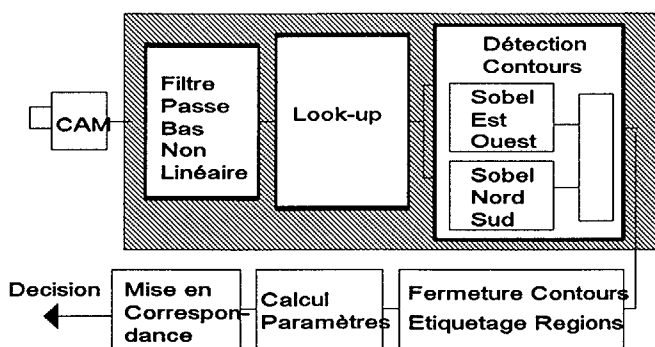


Figure 4.1

4.1 Les Algorithmes

L'algorithme proposé du filtre médian pour des fenêtres 3x3, est un filtre médian séparable [Nar 85]. Pour une image $I(m,n)$, la valeur $O(m,n)$ d'un pixel traité est donnée par :

$$O(m,n) = \text{median}[\text{median}[I(k,n-1), I(k,n), I(k,n+1)]]$$

où: $k=m-1, m, m+1$, et m et $n=1, \dots, L$. L est taille de l'image. Cet opérateur de base exige 6 opérations élémentaires de comparaison/échange.

Le convertisseur linéaire-logarithme est mis en oeuvre par l'application d'une table de look-up.

L'opérateur de Sobel est utilisé pour générer l'image gradient avec les contours potentiels. L'algorithme correspond à la convolution de l'image par les masques suivants.

$$\begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array} \quad \begin{array}{ccc} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{array}$$

Sobel Nord-Sud

Sobel Est-Ouest

Le post-traitement de l'opérateur de Sobel est le maximum des valeurs absolues des deux réponses Nord-Sud et Est-Ouest. Cette approche exige 12 additions soustractions et une comparaison/échange.

4.2 Les Architectures

Les architectures proposées pour les algorithmes sont montrées dans la Figure 4.2. Le filtre médian est composé d'un pipeline de deux trieurs monodimensionnels à trois entrées. Le médian vertical trie les trois pixels d'une colonne d'une fenêtre 3x3 de l'image d'entrée et fournit le médian. A chaque cycle d'horloge, le médian horizontal calcule le médian des médians des trois colonnes.

La table de look-up de l'égalisation d'histogramme est réalisée par une mémoire ROM de 256 octets, configurée avec les valeurs de niveaux de gris de la courbe linéaire-logarithme de transformation. Le pixel de l'image d'entrée est utilisé comme adresse de la mémoire.

L'architecture proposée pour la réalisation des deux convolutions de l'opérateur de Sobel est un pipeline d'additionneurs et soustracteurs, résultat d'une adaptation algorithmique basée sur la séparabilité et le codage en puissance de 2 des coefficients du filtre (Figure 4.2). En effet, la séparabilité, comme pour le cas du médian, réduit le filtre à l'application successive de deux filtres monodimensionnels (filtres $[1 \ 0 \ -1]$ et $[1 \ 2 \ 1]$), aux colonnes et aux lignes de trois pixels de la fenêtre 3x3 de l'image d'entrée. Le codage en puissance de 2 des coefficients réduit la multiplication par deux à un décalage d'un bit en entrée des additionneurs, dans les points indiqués par un cercle dans la Figure 4.2.

4.3 Estimation de surface et vitesse

A partir de l'application des équations 3.2, 3.3, et en considérant un XC4010-5, nous obtenons les valeurs montrées dans le tableau 4.2 pour la surface (Arch) et vitesse (Tc) :

L'architecture de la chaîne complète de pré-traitement prend donc 238 CLBs, ce qui correspond à une utilisation de 60% de la surface effective d'un PGA. 72 IOBs sont nécessaires pour les communications avec les mémoires FIFO externes (50% des ressources d'entrées/sorties disponibles). La fréquence F_{max} maximale interne permise est donnée par $F_{max} = 1/49 \text{ ns} = 20 \text{ MHz}$

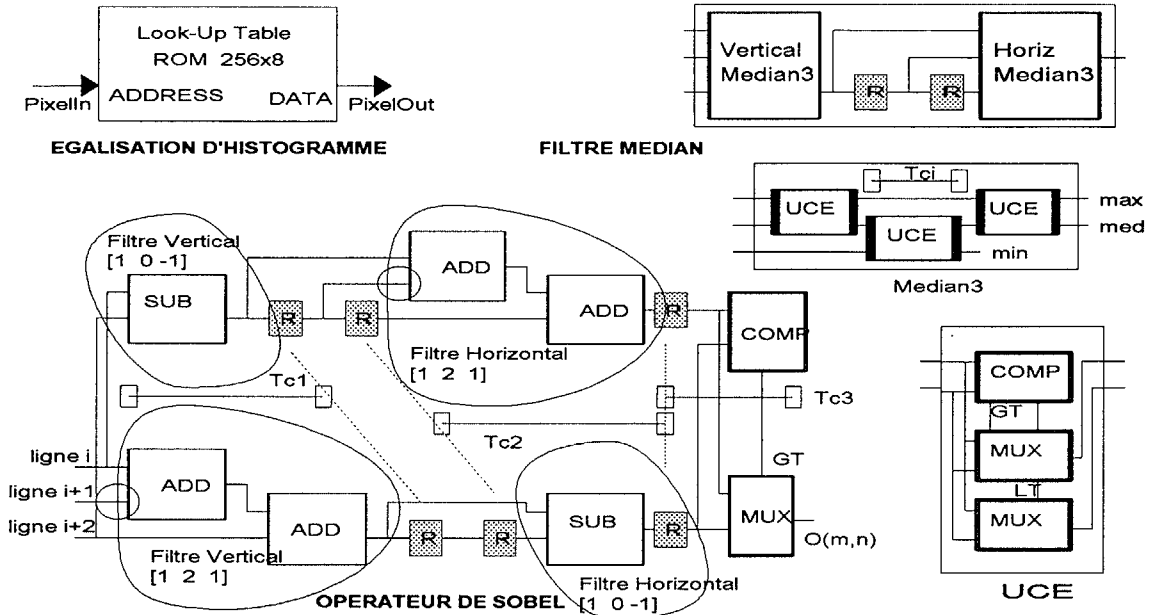


Figure 4.2 - Architectures du Module d'égalisation d'histogramme, des filtres Médian et Sobel.

	Médian Séparable	Egalis d'Histogramme	Opérateur de Sobel
Arch	$\sqrt{N}(\sqrt{N}-1)S_{UCE} + 6S_{RGB} + 0 + 0$ =6.13+6.4=102 CLBs	$N_{OCT}/4+S_{dec}$ =256/4+5=69 CLBs	$(6.S_{ADD}+S_{UCE})+6.S_{REG8}$ =(6.5+13)+24=67 CLBs
Tc	$N_{UCE}(T_{INT}+T_{UCE})+3$ =1(5+5T _{CLB})+3=38 ns	$N_{CLB}(T_{INT}+T_{CLB})+3$ =3(11)+3=35 ns	MAX[Tc1;Tc2;Tc3]=49ns Tc1=1(T _{ADD} +T _{INT})+3=3.T _{CLB} +5+3=26ns Tc2=2(T _{ADD} +T _{INT})+3=6.T _{CLB} +10+3=49ns Tc3=N _{UCE} (T _{INT} +T _{UCE})+3 =1(5+5T _{CLB})+3=38 ns

Tableau 4.2

5. Conclusions

Nous avons présenté un modèle d'architectures adapté aussi bien aux algorithmes de traitement bas niveau d'images qu'à leur implémentation sur des circuits FPGA. Ce modèle est basé sur un traitement en flot de données et structuré sous forme d'un pipeline d'opérateurs de base.

La méthode d'évaluation proposée permet l'étude de faisabilité des algorithmes sur ce type de circuit, ainsi que la détection des points critiques dans l'architecture (étages de pipeline), lesquels nécessitent d'être décomposés en sous-opérateurs de base.

Nous montrons que l'implémentation efficace d'un algorithme de traitement bas niveau d'images sur des FPGAs exige la prise en compte des particularités architecturales des circuits, et que dans tel cas, les circuits FPGA Xilinx sont adaptés pour la réalisation d'applications de traitement à la fréquence vidéo.

6. Bibliographie

[Alves 92] M. ALVES DE BARROS, M. AKIL. "Vers la synthèse d'architectures pour le traitement bas niveau d'images: un exemple de réalisation sur technologie reconfigurable".

Annales de la 1^{ère} Conférence Adequation Algorithmes-Architectures", Lannion, France, Septembre 1992, 67-72.

[Filloq 91] J. M. FILLOQUE, E. GAUTRIN, B. POTTIER. "Efficient computation on processor networks with programmable logic", PARLE91 proceedings, Springer-Verlag, 1991.

[Wes 85] N. WESTE, K. ESHRAGHIAN *Principles of CMOS VLSI design: A System Perspective*. Addison Wesley Publishing Company, 1985.

[Xil 92] *The XC4000 Programmable Gate Array Data Book*, Xilinx, San Jose, CA, USA, 1992.

[Nar 85] P. M. NARENDA "A separable median filter for image noise smoothing", In: *Digital Image Processing and Analysis*. Vol 1. Digital Image Processing, IEEE Computer Society, 1985, 450-459.

[Ruetz 86] P. Ruetz, R. Jain, C Shunc, J. Rabast, J. Jacobs, R. Broderon. "Automatic layout generation of real time digital image processing circuits". *Proceedings of the Custom Integrated Circuits Conference*. 1986, 111-115.