

SYNTHESE D'ARCHITECTURE ET MULTIPROCESSEUR HETEROGENE: UNE APPLICATION EN TRAITEMENT D'IMAGES.

PHILIPPE JL, MARTIN E DUBOIS H, SENTIEYS O.

Laboratoire d'Analyse des Systèmes de Traitement de l'Information,
ENSSAT 22300 LANNION, FRANCE; Tel: (33)96/46/50/30; FAX: (33)96/37/01/99;

RÉSUMÉ

Les applications de traitement du signal et de l'image sous contraintes de temps réel d'exécution nécessitent des moyens de calcul qui exploitent de plus en plus souvent le parallélisme. Il est alors nécessaire de concevoir les architectures de tels systèmes avec méthode, si l'on veut maîtriser leur coût de développement. Nous proposons une méthode de conception d'architectures MIMD hétérogènes, fonctionnant en mode SPMD. Ces architectures sont conçues à l'aide de deux démarches complémentaires: l'analyse prévisionnelle de performances et la synthèse architecturale. Cette approche a été expertisée sur une application de restauration d'images floues, par filtrage de Kalman 2D.

ABSTRACT

Applications in signal processing under real time constraints require parallel architectures. In order to keep the implementation as efficient as possible, we need CAD tools dedicated to this application field. We present in this article a methodology depending on objective performance measures, and high level synthesis in order to design heterogeneous multi processors networks. It is applied to a Kalman filtering algorithm for images degraded by both blur and noise.

1 Introduction

La mise en oeuvre d'applications de traitement du signal et de l'image (TDSI) met en valeur deux points particuliers. Tout d'abord les contraintes temps réels souvent associées à ce domaine, imposent l'utilisation de structures d'exécution performantes. Celles ci doivent être le plus possible en adéquation avec l'algorithme, tout en étant suffisamment génériques pour pouvoir être réutilisées. En outre la maîtrise du coût et de la complexité de telles architectures, intégrant le plus souvent le parallélisme sous différentes formes, va imposer de plus en plus des outils méthodologiques permettant d'assister l'algorithmicien durant la phase de matérialisation. Pour répondre à ces deux points, nous proposons une démarche basée sur un modèle d'exécution appartenant à la classe des multiprocesseurs MIMD hétérogènes; le mode de parallélisme employé repose sur un partage des données à traiter (mode SPMD). La méthodologie mise en oeuvre intègre l'évaluation prévisionnelle de performances ainsi que la synthèse d'architectures.

L'analyse prévisionnelle aura pour charge de prédire les performances d'un réseau multiprocesseur pour un algorithme donné, en fonction du nombre de processeurs, de leur puissance à calculer et à communiquer, et de la topologie du réseau d'interconnexion.

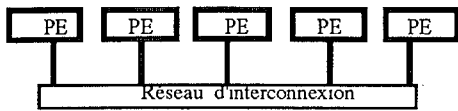
La synthèse d'architectures permettra de renforcer la puissance de calcul des processeurs élémentaires, en leur adjoignant des processeurs dédiés. L'association de ces deux méthodes vise à optimiser le coût total de l'architecture.

2 Méthodologie de conception

2.1 Architecture cible.

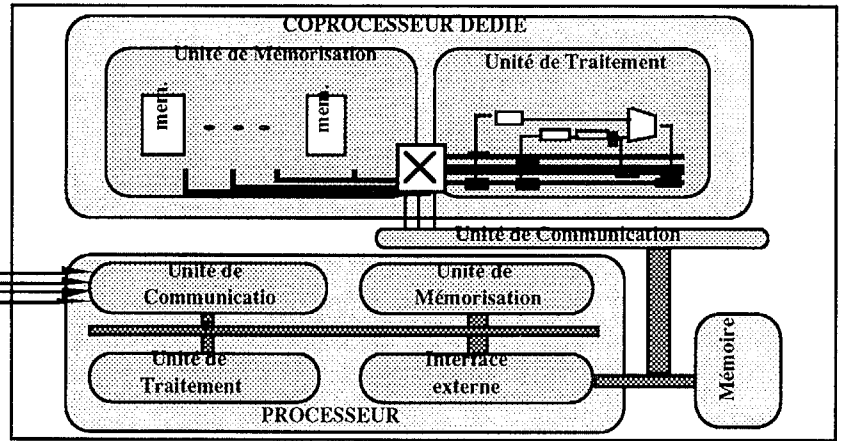
Pour répondre au critère d'optimalité et de versatilité des architectures, nous avons opté pour un modèle d'exécution MIMD hétérogène. La versatilité est atteinte par l'utilisation de processeurs de contrôle, et l'optimalité par l'adjonction de processeurs dédiés. Chaque processeur est donc constitué d'un processeur standard (type Transputer), et d'un circuit spécifique de calcul (ASIC) connecté sur son bus. Ce coprocesseur dédié repose sur un modèle coeur de processeur de traitement du signal (PTS).

Le processeur de contrôle est chargé d'alimenter l'ASIC en données à traiter et à communiquer à travers le réseau avec les autres processeurs. Ces communications auront pour but, avant le calcul, de diffuser les données, d'échanger des valeurs en cours de calcul, de centraliser les résultats en fin de calcul. L'ASIC matérialise la part de l'application jugée critique en terme de temps de calcul.



Communication avec les autres processeurs élémentaires

Modèle MIMD et organisation d'un processeur élémentaire, comprenant un processeur standard et un processeur spécialisé.



PROCESSEUR ELEMENTAIRE (PE)

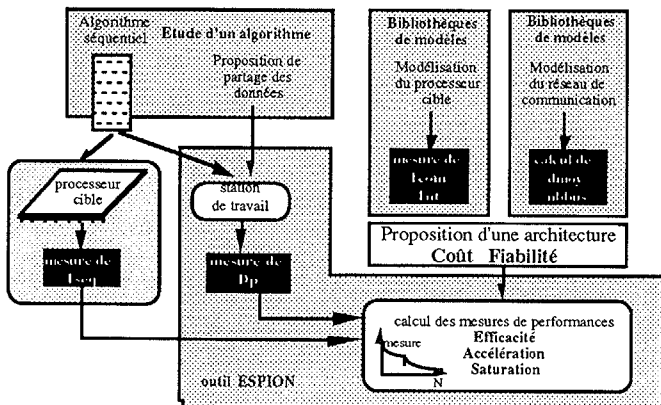
2.2 Analyse prévisionnelle de performances.

Celle ci consiste à prédire le temps d'exécution d'un algorithme parallélisé à partir d'une analyse de son code séquentiel. Cette prédiction repose sur une triple modélisation: processeur, réseau et algorithme.

Algorithme: p% (fraction des tâches parallèles)
Tseq (temps de calcul séquentiel)
Dp (nombre de communication entre deux partitions)
Réseau: dmo (distance moyenne entre 2 PE)
Nbbus (nombre de bus ou de liens)
Processeur: C (taux de communication)
Tcom (temps d'échange d'une donnée)
Tnt (taux de pollution des échanges sur les calculs)

Des mesures de performance ont été définies: efficacité, accélération, seuil de saturation. Cette dernière mesure permet de déterminer si le temps d'exécution parallèle d'un algorithme dépend du temps de traitement (les communications sont alors masquées, le réseau est hors saturation) ou du temps nécessaire à écouler les communications (les calculs sont masqués, le réseau est en saturation).

Ces différents paramètres seront calculés ou résulteront d'une analyse de l'exécution de l'algorithme séquentiel sur le processeur cible. La contrainte de coût nous amènera à rechercher le nombre de processeurs le plus faible associé autour d'un réseau le moins couteux. Cette méthode a été intégrée dans un outil, ESPION [DUB91].



Ainsi pour un algorithme de restauration d'images par filtrage de Kalman et un schéma de parallélisation [BIE83], les résultats suivants ont été obtenus [MAR91, SEN93].

N	1	2	4	8	10
temps prévu		2,013	1,037	0,568	0,482
temps mesuré	4,21	2,039	1,049	0,575	0,457
erreur de prédiction		1,3%	1,16%	1,23%	5,1%

2.3 Synthèse architecturale.

Celle ci a pour but, à partir d'une description comportementale d'un algorithme, d'une bibliothèque de ressources et d'un énoncé des contraintes, de synthétiser une architecture matérialisant l'algorithme et satisfaisant ces contraintes [AIR92]. Dans notre approche, les contraintes incluent le temps d'exécution de l'algorithme (contrainte de temps réel), ainsi que le coût du coprocesseur dédié. Cette architecture est composée de 4 unités fonctionnelles:

- * traitement des données: cette unité est organisée autour de cellules, chacune comprenant un opérateur de calcul et des registres.
- * mémorisation des données: cette unité est organisée autour de différents bancs mémoire, chacun disposant de son propre générateur d'adresse.
- * communication: à partir de FIFO, elle permet d'assurer la communication avec le processeur hôte.
- * contrôle.

L'outil développé GAUT¹ [MAR93], a comme point d'entrée un algorithme de TDSI décrit en VHDL comportemental et fournit actuellement la description structurelle de l'unité de traitement, et celle de l'unité de contrôle sous la forme d'un automate à états finis.

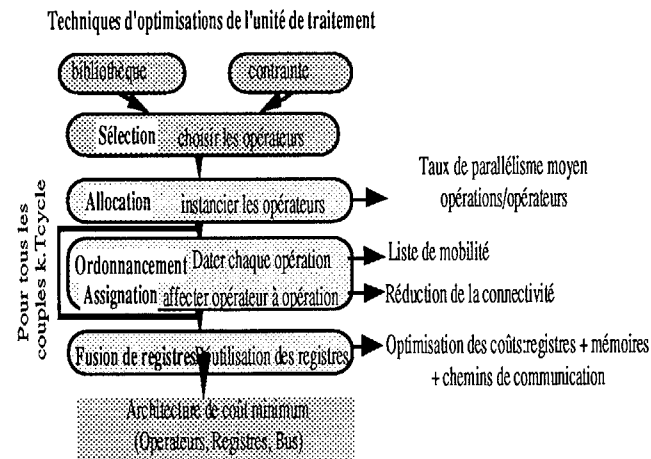
La bibliothèque formelle contient des composants primitifs (additionneur, registres, ...) ainsi que le résultat d'anciennes

¹GAUT est supporté par une convention du CNET de Meylan



synthèses. Un composant de la bibliothèque est modélisé à l'aide de deux paramètres représentant, l'un son temps de fonctionnement, et l'autre son coût formel qui, par exemple, peut être lié à la surface du composant.

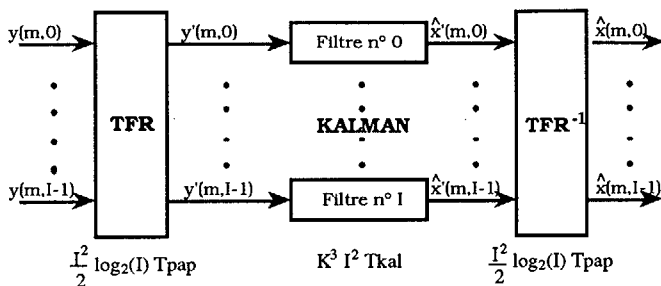
Lors de la synthèse GAUT sélectionnera donc les composants en nombre juste suffisant pour satisfaire les contraintes de temps et de minimisation du coût de l'architecture. Outre la synthèse de coprocesseurs dédiés, la synthèse d'architectures permet d'explorer avec méthode l'espace des solutions architecturales pour un algorithme de TDSI donné [PHI93]. En effet de nombreuses synthèses peuvent être itérées après avoir changé la bibliothèque, voir les contraintes temporelles. Cette démarche permet de prototyper très rapidement l'implantation d'un algorithme. A cet effet diverses informations relatives à l'occupation temporelle des différentes ressources permettent de sélectionner l'architecture du VLSI le plus en adéquation avec l'algorithme à implanter. En outre l'analyse de la complexité de l'architecture ainsi conçue fournit une estimation originale de la complexité de l'algorithme (analyse à rapprocher de l'approche traditionnelle consistant à compter le nombre d'opérations). L'organisation de l'outil est présentée ci dessous.



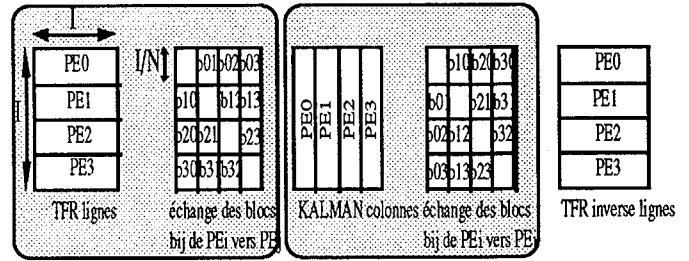
3 Résultats.

L'analyse et la synthèse ont été utilisées dans le cadre de la conception d'une architecture hétérogène pour la restauration d'image par filtrage de Kalman 2D. La synthèse permet de définir un coprocesseur dédié afin d'augmenter la capacité de traitement du processeur élémentaire. A capacité de traitement donné, l'analyse permet de sélectionner l'architecture multiprocesseur la moins couteuse.

Le filtrage de Kalman peut être parallélisé dans le domaine de Fourier [BIE83], ce qui conduira à la représentation suivante:



La mise en oeuvre parallèle se fait par partage de l'image en lignes (pour les FFT), ou en colonnes (pour les filtres de Kalman). Les processeurs effectuent la rotation de l'image par des communications masquées par les calculs [SEN93].



Considérons une cadence de 20 images par seconde et des images ayant une définition de 256x256 pixels. Une analyse prévisionnelle nous a montré qu'une telle performance ne pouvait être atteinte à partir des processeurs standards (prévision de 0,6s pour un anneau de 4 TMS320C25, de 0,4 s pour un anneau de 32 Transputer. Le respect de la contrainte temps réel nous a amenés à concevoir une architecture hétérogène basée sur des associations processeurs de contrôle/ coprocesseur de traitement.

Le circuit spécialisé est chargé d'exécuter le calcul sur un point. Le processeur de contrôle est chargé de l'alimenter à une cadence imposée. L'alimentation en données conduira les différents processeurs de contrôle à échanger des données entre eux (rotation de l'image).

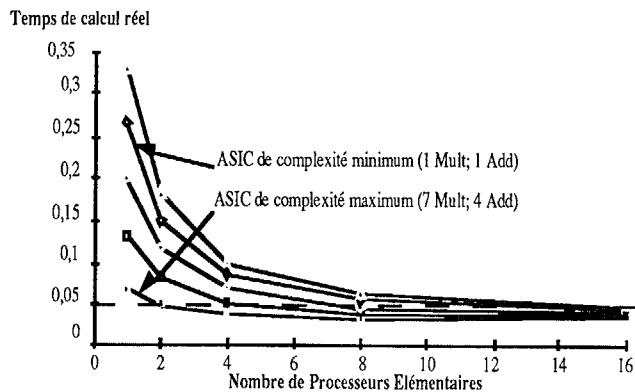
Les fichiers traités par GAUT sont représentés sur la figure suivante (t_lat représente la cadence d'arrivée des échantillons, c.a.d la contrainte temps réel imposée à l'outil).

contenu de la librairie	spécification de l'algorithme
PACKAGE operateurs IS	ENTITY kalman ISPORT
CONSTANT t_lat: integer:=	(...)
400;	END kalman;
CONSTANT t_add: integer:=	ARCHITECTURE COMP
25;	OF kalman IS
...	-- déclarations
COMPONENT registre (...);	BEGIN PROCESS (ymr)
....	--déclarations- BEGIN
COMPONENT add_sous	--description de l'algorithme
GENERIC
(surface:integre:=15;	Z=AZ+Ks(y - CCZ)
function:string="add,sub";	X=Z+E
entree:integre:=2);	...
PORT (...);	END PROCESS.
END component;	END comp;
...	
END operateurs	

Sur le schéma page suivante nous représentons le résultat de la synthèse pour le filtrage de Kalman (description de l'unité de traitement).



Nous avons effectué la synthèse de plusieurs ASICs. Les résultats sont donnés sur le graphe suivant:

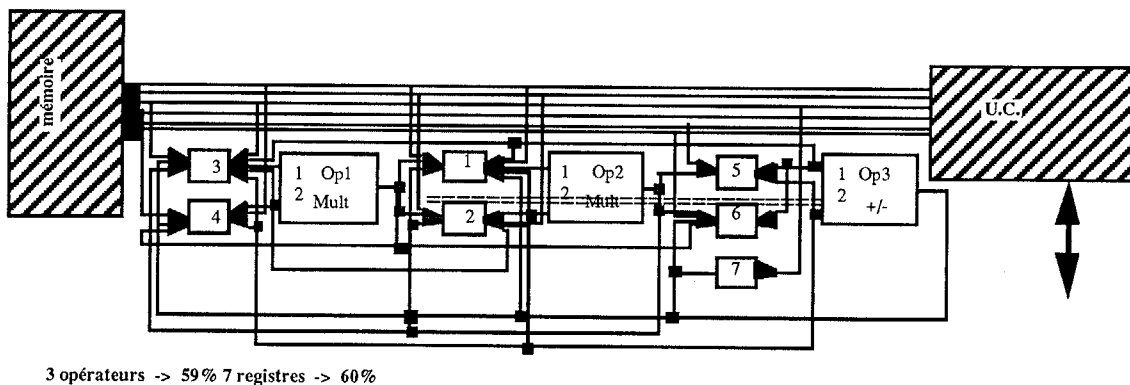


On constatera ainsi que la mise en oeuvre en temps réel de l'application peut se faire à l'aide d'un réseau de 2, 4, 8 ou 16 Transputers, associés respectivement à des ASICs de complexité (7mult,4 add), (2mult+2add), (2mult+2add) et (1mult+1add). La décision finale revient à l'utilisateur, qui doit

pondérer au mieux le coût global du système qui intègre le coût du processeur dédié mais aussi celui lié au nombre de processeurs et au réseau d'interconnexion.

4 conclusion.

Notre approche vise à définir un environnement de CAO d'architectures hétérogènes en utilisant un outil d'aide au niveau système et un outil de CAO de VLSI au niveau composant. L'intégration de ces deux démarches nous permet de concilier l'approche machine reprogrammable et dédiée tout en permettant de garder notre méthodologie initiale de choix d'une structure multiprocesseur basée sur l'évaluation de performances. Cette approche permet d'explorer avec méthode l'espace de recherche des solutions, et décharge le concepteur des tâches les plus répétitives. Nous travaillons actuellement à formaliser une stratégie de conception, qui permette un balancement optimal des coûts de l'architecture hétérogène entre celui des ASICs et celui du réseau.



BIBLIOGRAPHIE:

- [AIR92] Airiau R. " Synthèse d'algorithmes à partir de VHDL", Conférence Adéquation Algorithmes Architectures, Lannion 14- 15 Septembre 1992.
- [BIEM 83] BIEMOND, J "A Fast Kalman Filter for Images Degraded by both Blur and Noise" *IEEE Transactions on Acoustics, speech and signal processing*. Vol 31 N°5 pp1248-1256 October 1983.
- [DUB91] Dubois, H "Configurable architectures: a methodologic approach applied to a transputer based machine for image processing". 7th Workshop on multidimensional signal processing Lake Placid 23-25 septembre 1991.
- [MAR91] Martin E. "Definition de mesures objectives de performances pour le mise en oeuvre parallèle d'algorithmes de traitement d'image" 13° colloque GRETSI Juan les Pins 16-20 septembre 1991.
- [MAR93] Martin E. "Traitement du signal et architecture dédiée, GAUT une approche méthodologique en CAO de VLSI" Congrès AFCET 93

- [PHI93] Philippe, JL "Prototyping digital signal processing using VHDL and CAD architectural tool", SPRING93 Innsbruck, 14,17 Mars 1993.
- [SEN93] SENTIEYS, O. "Application de l'outil ESPION pour l'analyse des architectures multiprocesseurs au filtrage de Kalman 2-D rapide" *revue Traitement du Signal* Vol 10 n° 1.