

Symphonie calculateur massivement parallèle : modélisation et réalisation

Symphonie Massively Parallel Computer: Modelling and Design

par **Thierry COLLETTE, Christian GAMRAT, Didier JUVIN, Jean-François LARUE,**
Laurent LETELLIER, Marc PEYTHIEUX, Renaud SCHMIT, Marc VIALA

LETI (CEA - Technologies Avancées)
CEA - DEIN- SLA
F-91191 Gif sur Yvette

résumé et mots clés

L'objectif de cet article est de présenter le calculateur embarqué de traitement d'image SYMPHONIE, ainsi que la méthodologie mise en oeuvre pour sa réalisation. Ce calculateur dont une des applications se situe au coeur du système de veille infrarouge de l'avion Rafale pourra comporter jusqu'à 1024 processeurs. Afin de tenir compte des contraintes de volume et de consommation, un ASIC d'un million de portes a été développé. Pour réussir cette réalisation un modèle VHDL a été écrit permettant des simulations de l'ensemble du système. Par ailleurs, ce modèle a permis d'aborder la réalisation du circuit ASIC en faisant appel aux outils de synthèse VHDL. Nous terminerons cette présentation par quelques performances dans différents domaines d'applications.

Traitement d'images, SIMD, ASIC, VHDL

abstract and key words

The goal of this paper is to present an embedded calculator for image processing SYMPHONIE and the methodology used for its realization. In this calculator we could have up to 1024 processors. Its first application will take place in the french Rafale Aircraft as core of the infra-red system. For this application, low size and low consumption will be very important and an ASIC of a million gates has been developed. To succeed in this realization we used a VHDL model allowing simulations on the full system. Then we used VHDL synthesis methods for the conception of the ASIC. We will conclude this paper by a presentation of some performances of the system in some applications fields.

Image processing, SIMD, ASIC, VHDL

1. présentation de l'architecture

Le nouveau concept Symphonie (*SYstème Massivement Parallèle à Haut Niveau d'Intercommunication Embarqué*), a pour objectif d'atteindre des performances élevées en traitement d'images tout en répondant aux contraintes sévères des systèmes électroniques embarqués. Auparavant les systèmes utilisés faisaient presque toujours appel à des circuits dédiés, solution alliant le niveau de performance requis à un encombrement réduit. Aujourd'hui, les

applications utilisent souvent des enchaînements complexes d'algorithmes très divers, ce qui nécessite de plus en plus de faire appel à des systèmes évolutifs. Cette évolution est confortée par le souci de rentabilité qui exige des systèmes programmables et donc réutilisables pour des applications différentes. Pour réaliser le coeur du système de veille infrarouge du Rafale, la SAT (Société Anonyme de Télécommunications) a choisi de développer le calculateur parallèle Symphonie en collaboration avec le LETI (Laboratoire d'Electronique du Commissariat à l'Energie Atomique).

Ce calculateur parallèle de type SIMD peut intégrer de 32 à 1024 processeurs super scalaires organisés en anneau (Fig. 1). La structure des processeurs, les différents liens de communication inter-

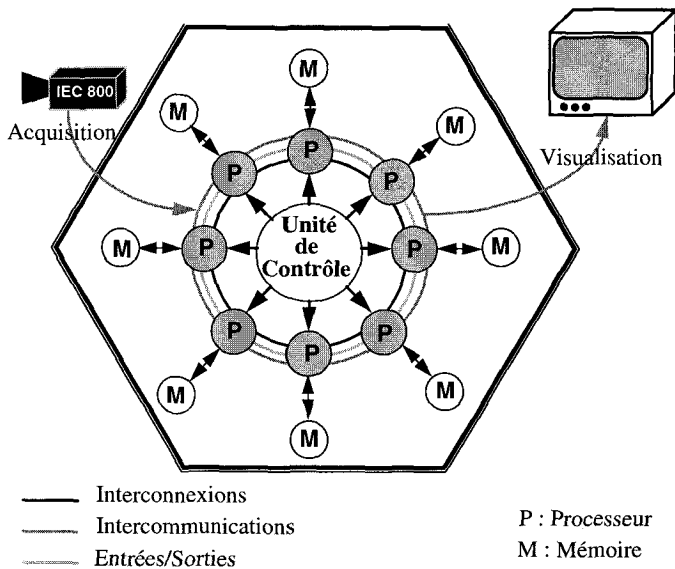


Figure 1. - Structure générale de Symphonie : une unité de contrôle distribue les instructions aux processeurs reliés entre eux par trois canaux de communication.

processeurs assurent un excellent taux de parallélisme. Grâce à ces caractéristiques un système à 1024 processeurs atteint une puissance effective de 50 Gops et 2,5 Gflops.

Pour atteindre le niveau de performances visé plusieurs concepts innovants ont été mis en oeuvre parmi lesquels :

- un puissant coprocesseur d'adressage par processeur, associé à une organisation originale des données [Ess 88], [Juv 88], permettant un accès en parallèle sans conflit d'accès ni effet de bord, à tous types de données bi-dimensionnelles. Le coprocesseur d'adressage décharge le processeur de tous les calculs d'adresses et permet l'accès aussi bien à des structures 2D régulières qu'à des structures 1D symboliques. Le processeur n'effectue plus alors que les opérations nécessaires au calcul algorithmique ce qui contribue à l'obtention d'une puissance effective de calcul proche de la puissance crête.

- un réseau d'intercommunication autonome basé sur un concept breveté [Col 94], [Let 93]. Ce réseau assure les échanges de données entre les processeurs, ces échanges s'effectuant en parallèle avec le traitement. Le format des données est de 32 bits et les échanges peuvent s'effectuer de façon irrégulière entre des processeurs quelconques. Des modes spécifiques permettent la globalisation des données ainsi que les communications régulières. Ce réseau vient compléter le réseau d'interconnexion, lui aussi sur 32 bits, qui ne permet que des échanges réguliers entre processeurs voisins.

- une réduction de l'encombrement par l'utilisation de MCM (Multi-Chip Module) et d'une technologie 0,5 µm de SGS Thomson. Quatre processeurs et 64 KOctets de mémoire tiennent sur une seule puce et quatre puces sont intégrées par MCM. Seize processeurs et 256 KOctets de mémoire sont ainsi intégrés dans

un seul boîtier. Une simple carte compact-PCI peut ainsi accueillir un système complet comprenant 32 processeurs et l'unité de commande. Cette dernière ayant également donné lieu à la réalisation d'un ASIC n'occupe qu'une place limitée.

2. les points forts de l'architecture

Les efforts de conception ont porté particulièrement sur les processeurs, les coprocesseurs d'adressage et le système d'intercommunication.

- Chaque processeur dispose d'une unité arithmétique et logique de 32 bits et d'un multiplieur 16 bits par 16 bits avec résultat sur 32 bits capables de fonctionner simultanément ainsi que d'un banc de 10 registres 32 bits à double accès. Il est ainsi possible de réaliser en un seul cycle de 50 ns une opération ALU, une multiplication et de sauvegarder les deux résultats. Le calcul flottant est émulé car l'intégration d'une vraie unité de calcul flottant aurait réduit le nombre de processeurs par ASIC (et donc augmenté le volume du système) et augmenté la consommation alors que le calcul flottant n'est pas majoritairement utilisé en traitement d'image. Toutefois des fonctions câblées, intégrées au processeur, accélèrent cette émulation d'un facteur 10.

- Le coprocesseur d'adresses associé à chaque processeur prend en charge la gestion de tableaux de données mono-dimensionnels ou bi-dimensionnels. Cette fonctionnalité facilite le stockage en mémoire et le traitement des primitives (formes, contours, couleurs...) extraites de l'image analysée. Le coprocesseur prend également en charge le balayage des images à traiter. La mise en oeuvre de ces fonctionnalités ainsi que l'organisation spécifique des données de type hélicoïdal impose un calcul d'adresse spécifique mettant en oeuvre plusieurs additionneurs ainsi qu'un certain nombre de registres de configuration.

- Le système de communication de Symphonie repose sur un anneau de cellules reliées entre elles par des liens 32 bits qui forment une boucle à décalages. Chaque cellule est connectée à un processeur. Elle a pour rôle d'insérer dans le réseau les messages du processeur auquel elle est associée et d'identifier parmi les messages qui circulent, ceux à destination de ce même processeur. Trois modes de communication sont possibles :

- le mode régulier (fig. 2) : Il permet d'échanger des données entre processeurs séparés par une même distance (par exemple distance 2);

- le mode globalisation (fig. 3) : Il permet à chaque processeur d'envoyer une donnée à tous les autres processeurs;

- le mode irrégulier (fig. 4) : il s'agit du mode le plus évolué. Il permet à chaque processeur d'envoyer un message constitué de 1 à 127 mots de 32 bits vers n'importe quel processeur identifié

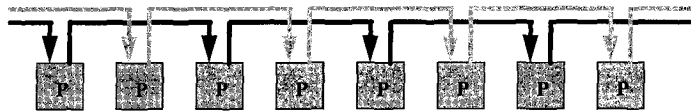


Figure 2. – Mode régulier.



Figure 3. – Mode globalisation.

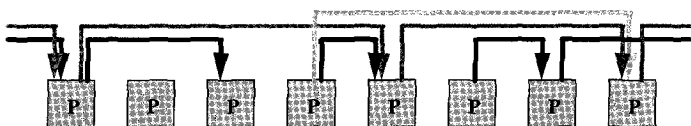


Figure 4. – Mode irrégulier.

par son adresse. L'émission et la réception des messages sont assurées par des FIFOs présentes dans chaque cellule. Le temps de traversée du réseau est proportionnel à la distance séparant le processeur émetteur et le processeur destinataire et dépend également du taux d'occupation du réseau.

– le mode mémoire : il s'agit d'une extension du mode irrégulier qui permet d'envoyer les données directement dans la mémoire du processeur destinataire sans passer par la FIFO de réception. Par contre il est nécessaire d'envoyer également sur le réseau l'adresse mémoire de destination. Comme la mémoire est sur 16 bits, cette adresse est codée sur les 16 bits de poids fort de la donnée.

3. méthodologie de conception

Afin de réduire au maximum l'encombrement et la consommation du calculateur SYMPHONIE plusieurs choix ont été effectués. Citons tout d'abord le choix de l'architecture SIMD qui optimise au mieux l'utilisation du silicium. D'autre part deux choix d'ordre technologique ont été effectués :

- utilisation d'une technologie 0.5 μm permettant d'intégrer quatre processeurs et 64 KOctets de mémoire dans un seul ASIC;
- intégration de quatre ASICs dans des modules multi-puces (MCM).

Ces différents choix nous ont permis d'atteindre les objectifs d'encombrement et de consommation attendus avec ce calculateur.

D'autre part pour concevoir l'ASIC, élément de base de la structure, nous avons choisi une méthodologie de conception assurant à la fois des possibilités importantes de validation de

l'architecture tout en apportant un gain significatif sur la durée de la phase de conception.

• Méthodologie de conception retenue

La méthodologie de conception choisie et mise en oeuvre dans ce projet, s'articule autour de la modélisation, de la simulation et de la synthèse VHDL. Elle se décompose en 3 phases fondamentales, qui sont :

- l'écriture et la validation de modèles VHDL synthétisables;
- la simulation VHDL globale du système avec des algorithmes significatifs;
- la conception de l'ASIC par synthèse VHDL.

Pendant la première phase, des modèles VHDL sont élaborés et validés à l'aide de programmes de tests fonctionnels. Ces modèles VHDL décrivent des blocs élémentaires du circuit définis lors d'un découpage préliminaire du système (Exemples : le coprocesseur d'adresse, le bloc de calcul...). Une fois écrits, ces modèles sont validés séparément à l'aide de programmes de tests fonctionnels spécifiques à chaque bloc élémentaire. Une fois tous les modèles écrits ils sont interconnectés entre eux de façon à obtenir un modèle de l'ensemble du système. Il est alors possible dans une deuxième phase de simuler l'ensemble du système à l'aide de programmes mettant en oeuvre des algorithmes significatifs (Exemple : filtre de Sobel...). Ces simulations permettent de vérifier la fonctionnalité de l'ensemble du calculateur. La troisième phase met en oeuvre les outils de synthèse VHDL qui permettent à partir des modèles VHDL synthétisables d'aboutir à une représentation au niveau portes élémentaires du circuit à réaliser. En pratique certains modèles peuvent être partiellement réécrits. En effet la synthèse peut dans certains cas aboutir à des représentations matérielles inadaptées et il est alors nécessaire d'être plus directif dans l'écriture du code VHDL. Il est alors nécessaire de revalider le modèle modifié par une nouvelle simulation au niveau système.

• Choix de la technologie

Afin d'optimiser au mieux le rapport performances/volume électronique, deux types d'intégration ont été utilisées pour ce projet. Tout d'abord, l'intégration ASIC avec la technologie HC-MOS 5 (0.5 μm) de SGS Thomson à partir de cellules standards (« standard cell »), qui offre les avantages suivants :

- pérennité du circuit, lié au choix d'une filière technologique récente. La technologie 0.5 μm n'est disponible que depuis 1995;
- meilleur rapport performances/rapidité de conception permis par la réalisation d'un circuit à partir de cellules standards (« standard cell »).

Ensuite, l'intégration sur module multi-puces qui revient à rapporter directement des puces nues sur un substrat offre les avantages suivants :

- augmentation de la fréquence d'horloge du système car les capacités de connexion entre puces sont plus faibles dans un module MCM que sur une carte traditionnelle;
- diminution de l'encombrement du système [Mas 94].

• Résultats et perspectives

Le flot de conception utilisé s'appuie sur deux chaînes de CAO, MENTOR et COMPASS. A l'origine, seuls les outils MENTOR devaient être utilisés, mais la technologie retenue n'était pas encore supportée par l'outil de synthèse de MENTOR. Ainsi les opérations de synthèse, d'optimisation et de vérification de timings ont été effectuées sur la chaîne COMPASS alors que toutes les opérations de modélisation VHDL et de simulation VHDL ont été effectuées sur la chaîne MENTOR avec les outils « Design Architect » et « Quicksim2 ».

La modélisation VHDL retenue pour ce projet, modélisation au niveau blocs synthétisables, nous a conduit à élaborer un modèle du système global fortement hiérarchique. Les simulations d'algorithmes de traitement d'images sur ce modèle étaient par conséquent très lentes. En contre partie, les opérations de synthèse logique se sont effectuées rapidement. En effet, seules des modifications mineures sont intervenues au niveau des modèles VHDL. De plus, ce type de modélisation VHDL nous a permis de développer rapidement le modèle du système complet. Ainsi, l'écriture des différents blocs VHDL a été effectuée par plusieurs personnes en parallèle.

Les opérations de synthèse logique sur la bibliothèque cible ne se sont effectuées qu'une fois la fonctionnalité du système bien établie et éprouvée par des simulations. Malgré quelques opérations de synthèses effectuées sur une autre technologie pendant la modélisation VHDL, certaines caractéristiques des modèles VHDL initiaux ont été abandonnées suite aux premiers résultats de synthèse et de placement.

Afin de garder cohérent le flot de conception, des passerelles depuis l'environnement MENTOR vers celui de COMPASS – et vice versa – ont été élaborées. Le transfert descendant (de conception) des schématiques et des modèles comportementaux s'est effectué grâce aux formats standards EDIF et VHDL. Seuls les vecteurs de test ont nécessité le développement d'une passerelle sophistiquée. Le transfert montant (de vérification) fut plus délicat. En effet, il s'agissait de rétro-annoter le modèle global du système avec les modèles synthétisés. Or la bibliothèque de synthèse n'existant pas sous MENTOR, seuls les modèles VHDL structurels au niveau portes pouvaient être utilisés afin d'effectuer ce transfert montant. Malheureusement, le simulateur « Quicksim2 » n'est pas complètement à la norme VHDL, si bien qu'il lui était impossible de compiler les résultats de synthèse. Il a donc été nécessaire d'employer un autre simulateur VHDL pour effectuer cette opération.

Le circuit ASIC conçu intègre 4 processeurs élémentaires et un bloc de décodage. Il comporte 250 000 portes équivalentes et 64 Koctets de mémoire SRAM. Le nombre de broches total du circuit est de 520 pour une taille de 3,5 cm². Le « floorplan » (fig. 5) permet d'apercevoir la régularité de la structure avec la zone occupée par les processeurs et leur mémoire qui se trouve dupliquée quatre fois.

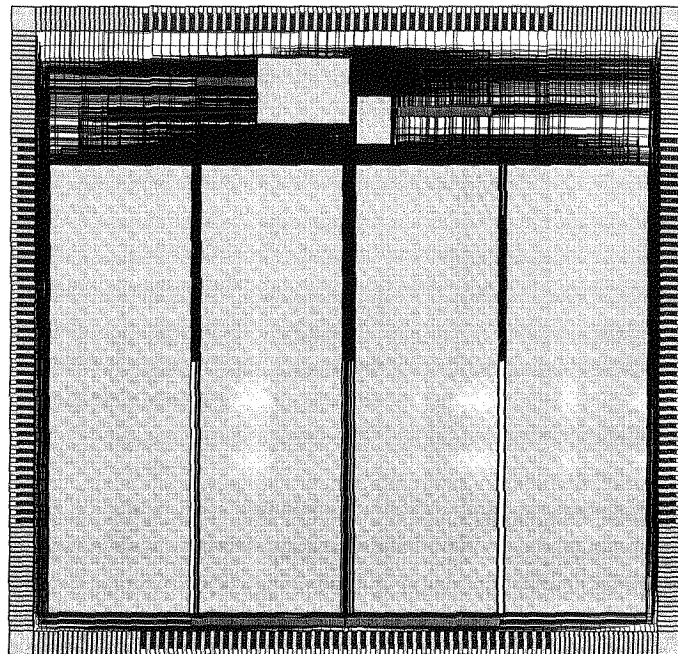


Figure 5. – «Floorplan» de l'ASIC Processeurs.

4. quelques applications de symphonie

S'agissant d'un calculateur développé principalement pour le traitement d'image, Symphonie montre en ce domaine des performances tout à fait remarquables. Néanmoins, certaines de ses particularités architecturales lui donnent accès à un spectre d'applications beaucoup plus large. Nous allons donc commencer par citer quelques résultats obtenus sur des algorithmes classiques de traitement d'images puis nous aborderons d'autres domaines dans lesquels le calculateur SYMPHONIE obtient des résultats intéressants. Nous évoquerons également les perspectives offertes par le couplage entre SYMPHONIE et des processeurs séquentiels.

• Traitement d'image bas niveau

Pour ces traitements l'architecture de Symphonie est particulièrement bien adaptée. Chaque processeur de la machine gère un sous ensemble des pixels composant l'image et permet de réaliser la plupart des algorithmes de traitement d'image à une cadence nettement supérieure à la cadence vidéo (Sobel, égalisation, étiquetage, etc.). Ceci permet d'envisager la mise en oeuvre d'une application complète (comportant un enchaînement significatif d'algorithmes de traitement d'images) en des temps parfaitement compatibles avec la cadence vidéo.

La disposition originale des données en mémoire (hélicoïdale) et le réseau d'intercommunication en anneau assurent ici une exploitation optimale des ressources de calcul (Puissance efficace = puissance crête). Cette bonne exploitation du parallélisme apparaît dans le tableau ci-dessous :

Tableau 1. – Temps de calcul sur des images 256x256 sur 8 bits.

	Symphonie 64 Prs	Symphonie 256 Prs
Convolution 3x3	0,98 ms	0,25 ms
Transformée de Hough	8,3 ms 64 directions	7,9 ms 256 directions
Egalisation d'histogramme	1 ms	0,35 ms

Une comparaison sur quelques algorithmes de base entre deux systèmes de coût et d'encombrement équivalents (c'est-à-dire une carte au format PCI) a été effectuée. Il s'agit d'une part de la carte GENESIS de Matrox basée sur le C80 de Texas Instrument associé à un processeur de voisinage (NOA) et d'autre part du calculateur Symphonie dans sa version minimale. Nous avons obtenu les résultats suivants :

Tableau 2. – Temps de calcul sur une image 512x512 sur 8 bits.

	C80	NOA	Symphonie 32
Convolution 5 x 5	34 ms	4,6 ms	18,3 ms
Histogramme	3 ms	Pas applicable	2 ms

• Réseaux de neurones

L'implémentation efficace de tels algorithmes requiert de nombreuses opérations de produit/accumulation (MAC). Ce sont typiquement les opérations de base dans de nombreux problèmes de traitement du signal. La capacité super scalaire des processeurs de Symphonie est ici utilisée de manière optimale ce qui lui donne des performances tout à fait honorables par comparaison à des machines spécifiquement conçues pour ce domaine telle MIND 1024 [Gam 93].

Tableau 3. – Performances en apprentissage (Hebb) et en relaxation pour un réseau de Hopfield de taille $N = 1024$.

Machine	Apprentissage	Relaxation
Sparc 20 70 Mhz	548 ms	108 ms
MIND-1024	143 ms	0,96 ms
Symphonie 128Prs	2.2 ms	2.0 ms

• Traitement du signal

Si les processeurs élémentaires de Symphonie sont essentiellement optimisés pour les calculs entiers sur 32bits, des fonctions câblées ont été intégrées afin d'accélérer notablement les calculs en virgule flottante. Ainsi des algorithmes nécessitant ce type de représentation des données peuvent être portés efficacement sur Symphonie. Par exemple, le filtre de Kalman, programmé sur un processeur de la machine donne des performances tout à fait attrayantes (tableau 4). Cette implémentation permet par exemple d'assurer en parallèle la poursuite de $N \cdot M$ objets différents (M étant le nombre de processeurs), chaque processeur appliquant le filtrage de Kalman aux trajectoires de N objets différents.

Tableau 4. – Performances (prédiction + estimation) d'une itération du filtrage de Kalman sur 1 seul processeur en fonction de la taille des vecteurs d'état (X) et de mesure (Z).

Dimension $X : Z$	3 : 2	4 : 2	5 : 2
Temps de calcul	0.47 ms	0.9 ms	1.57 ms

L'énoncé de ces quelques applications n'est certes pas limitatif et de nombreux autres domaines peuvent bénéficier des performances de Symphonie. En particulier toute application traitant un nombre important de données matricielles et présentant des contraintes temps-réel et/ou d'embarquabilité est un candidat naturel pour une implémentation efficace sur Symphonie.

Par ailleurs le développement de nouvelles applications sur Symphonie peut se faire dans des délais très raisonnables grâce aux outils de développement disponibles :

- le compilateur « SPL » (langage parallèle dédié) permet d'exploiter pleinement les possibilités de Symphonie par un accès direct aux fonctions de base tout en offrant l'agrément d'une syntaxe basée sur celle du « C »,

- le compilateur « C » (Développé à partir du compilateur gcc) permet de programmer en 'C'

Symphonie sans être trop pénalisé dans l'utilisation des fonctions de base grâce à une bibliothèque de fonctions spécifique (Gestion des communications, utilisation du coprocesseur d'adresses...),

- le compilateur « CT++ » (langage data-parallèle dérivé du C++ [Pic 95]) permet de développer du code parallèle sans connaître l'architecture de Symphonie,

- le simulateur/débogueur fonctionnel permet de développer et de déboguer une application sans disposer du calculateur et en bénéficiant de la convivialité d'une interface utilisateur sous X-WINDOWS.

5. intérêt du couplage avec un processeur de type sharc

Des travaux récents ont permis d'évaluer les performances d'une architecture hétérogène mettant en oeuvre un processeur SHARC [AD 95] associé à un SYMPHONIE 32 Processeurs sur une application réelle. Ce type d'architecture qui évoque le couplage C80-NOA de chez Matrox exploite pleinement la complémentarité des architectures SIMD entière de SYMPHONIE et séquentielle flottante du SHARC.

L'application évaluée consiste en une chaîne typique de vision industrielle ayant pour but de caractériser des objets se détachant sur un fond. La chaîne de traitements prend en entrée une images 8 bits de taille 256 par 256. Elle fournit en sortie une liste d'objets et leurs paramètres. L'application comporte deux phases principales. Une phase de pré-traitements, de nature pixélique, permet d'obtenir l'élimination du fond, le filtrage et la binarisation de l'image résultante. La phase d'analyse des objets, de nature symbolique, consiste en un étiquetage et un calcul des paramètres des objets.

L'association de la machine SYMPHONIE avec un processeur séquentiel performant a été étudiée. Le choix du SHARC par sa faible consommation conserve à la structure hétérogène la possibilité qu'a SYMPHONIE d'être embarquée. Le SHARC et SYMPHONIE sont étroitement couplés par une mémoire double port, accessible en parallèle par les processeurs de SYMPHONIE, et qui, vue du SHARC, se présente sous la forme d'une zone mémoire unique.

Tableau 5. - Temps d'exécution d'une application de caractérisation d'objets (en ms).

phase du traitement	SYMPHONIE 32	SHARC	SYMPHONIE 32-SHARC
pré-traitements	15.5	19.5	2.9
communication	-	-	0.9
analyse des objets	13.1	0.4	1.8
total	28.6	19.9	5.6

L'application présentée ci-dessus a été implantée sur simulateurs. Les temps d'exécution obtenus sont présentés dans le tableau 5. Ces résultats mettent en évidence le gain de temps obtenu par l'utilisation de cette architecture hétérogène, avec exécution des traitements pixéliques sur la machine SYMPHONIE et des traitements symboliques sur le SHARC. Les fonctions matérielles du SHARC prévues pour mettre en oeuvre le parallélisme apportent la possibilité d'une extension directe du nœud SYMPHONIE-SHARC en une architecture multi-SIMD constituée de plusieurs de ces nœuds. En effet, une architecture SIMD constituée d'un

grand nombre de processeurs (par exemple 512) manque de souplesse dès lors qu'il s'agit de traiter un nombre important d'éléments constitués eux mêmes de quelques dizaines de données (Cas d'imagettes extraites de l'image principale ou d'informations symboliques codées sous forme de vecteurs). Il est alors plus intéressant de disposer d'une architecture composée de plusieurs nœuds utilisant une structure SIMD réduite (par exemple 8 nœuds de 32 processeurs) qu'une large structure SIMD unique (par exemple 256 processeurs). Les échanges de données entre les différents nœuds Symphonie-SHARC sont essentiellement assurés par les liens de communication du Sharc. Au nombre de six, ceux-ci permettent de connecter les nœuds de calcul suivant une grande variété de topologies (grille 2D, grille 3D, anneaux, arbres, etc.). Cette caractéristique est importante puisqu'elle autorise le choix de la topologie la plus adaptée pour une application donnée. Le contrôle global est assuré par un SHARC maître.

6. conclusion

La méthodologie de conception retenue pour le projet SYMPHONIE a fortement accéléré les opérations de synthèse logique au détriment du temps de simulation. L'utilisation d'une chaîne de conception totalement intégrée facilite les différentes itérations intervenant lors de la conception d'un système. De plus, avec l'apparition des technologies sub-microniques, il apparaît que cette chaîne doit impérativement intégrer des outils de synthèse logique et de placement, ainsi que la bibliothèque cible.

L'organisation originale des données ainsi que le réseau d'intercommunications permet de traiter efficacement sur SYMPHONIE la totalité d'une application. Toutefois la parfaite adéquation de SYMPHONIE aux traitements pixéliques associée aux capacités de traitement symbolique d'un processeur SHARC permet d'atteindre des performances encore plus élevées sur l'ensemble d'une chaîne algorithmique. Ces performances peuvent être encore renforcées par l'utilisation d'une structure multi-SIMD mettant en oeuvre plusieurs nœuds SYMPHONIE-SHARC.

Le premier prototype du calculateur SYMPHONIE sera réalisé et testé début 1998.

BIBLIOGRAPHIE

- [AD 95] ADSP-2106X SHARC User's Manual, Analog Devices, First Edition, mar. 1995
- [Col 94] T. Collette, H. Essafi, D. Juvin, J. Kaiser : «*Sympati X : aSIMD computer performing the low and intermediate levels of image processing*» Future generation computer systems Vol 10 n°1, Avril 1994.
- [Ess 88] H. Essafi, «*Les processeurs ligne en traitement d'image*» : thèse de doctorat de l'université Paul Sabatier de Toulouse, 1988.
- [Gam 93] C. Gamrat, P. Peretto, «*MIND-1024 : A 1024 neurons fully connected real-time neurocomputer.*» Proc. WCNN93 Portland USA Vol 4 pp 783-786, 1993.

[Juv 88] D. Juvin, J.L. Basille, H. Essafi, J.Y. Latil, «*Sympati 2, a 1.5 D processor array for image application*», Signal processing IV : Theories and Applications, J.L. Lacoume, A. Chehikian, N. Martin, nad J. Malbos, Elsevier Science Publishers B.V. - North Holland, 1988.

[Let 93] L. Lettelier «*Synthèse d'images temps réel sur réseau linéaire de processeurs SIMD : algorithmes et architectures*» thèse de doctorat de l'Institut National Polytechnique de Toulouse - Toulouse, 1993.

[Mas 94] M. Massénat : «*MCM d'hier... et de demain*» Ed Polytechnica, 1994.

[Pic 95] M. Pic, H. Essafi, M. Viala, L. Nicolas «*T++ : a parallel object oriented language for a task and data parallel programming*» Computer Architectures for Machine Perception CAMP-95, Como, Italie - V. Cantoni L. Lombardi M. Mosconi M. Savini A. Setti, IEEE computer Society Press pp 216-220, 1995

Manuscrit reçu le 23 Juillet 1997.

LES AUTEURS

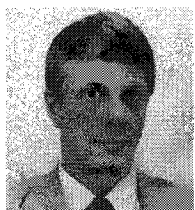
Thierry COLLETTE



Thierry Collette né en 1965, est ingénieur CUST en génie électrique (1988) ainsi que docteur en micro-électronique de l'Institut National Polytechnique de Grenoble (1992). Il est recruté au CEA-Leti (Direction des Technologies Avancées) en 1992 pour prendre la responsabilité de la modélisation VHDL et de la conception du calculateur SYMPHONIE. Depuis 1996, il est responsable de projet au sein du Leti. Il a présenté une vingtaine de communications à des conférences internationales, rédigé 3 publications et déposé

3 brevets. Ses principaux domaines de compétences sont la réalisation de systèmes de traitement de données temps réel, la conception de calculateurs très fortement intégrés ainsi que la réalisation d'ASIC en technologie avancées. Il encadre actuellement des travaux de recherche dans le domaine des interconnexions optiques appliquées aux calculateurs parallèles et celui du test et de la fiabilisation de ces calculateurs.

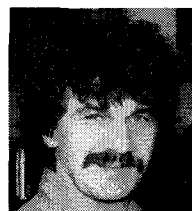
Didier JUVIN



Didier JUVIN né en 1956 est diplômé de l'Institut Supérieur d'Electronique de Paris en 1977, et possède une thèse de Docteur Ingénieur en Electronique depuis 1983 sur la reconnaissance d'objets 2D (système ANIMA). Recruté au CEA en 1983 pour transférer ses travaux de thèse à la société GIXI afin de constituer une famille de systèmes de vision industrielle (famille VX), il devient ensuite responsable du projet de calculateur massivement parallèle dédié au traitement d'images SYMPATI 2, mené en collaboration avec

l'IRIT de Toulouse, commercialisé par CENTRALP (machines OPENVISION). En 1988, il devient responsable du Groupe Architectures et Algorithmes Parallèles. Cette équipe développe des architectures innovantes de calculateurs parallèles, ainsi que des applications industrielles associées au traitement d'images. Depuis 1995, il est responsable du Service Logiciels et Architectures au sein du LETI. Il a 41 présentations à des conférences internationales, 3 publications et 6 brevets. Son principal domaine de compétences est dans le domaine du traitement d'images et des architectures de calculateurs parallèles, ainsi que le développement de structures électroniques complexes pour le traitement de données temps réel.

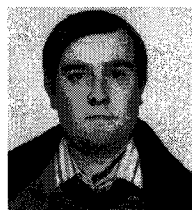
Christian GAMRAT



Christian Gamrat né en 1957 a Lyon, ingénieur diplômé de l'état, il est arrivé au CEA en 1982 au centre de Grenoble, il a d'abord participé à la conception de systèmes d'instrumentation et d'acquisition pour les expériences de physique nucléaires. En 1988, il a participé à la définition et à la conception de machines dédiées aux réseaux de neurones formels. C'est ainsi qu'en relation avec une équipe de physicien il a dirigé en 1990 la réalisation du calculateur neuronal MIND-1024. En 1994, il a rejoint

l'équipe d'architectures parallèles du CEA/Leti à Saclay au sein de laquelle il a participé au développement du calculateur Symphonie et s'intéresse aux architectures avancées pour le traitement d'images.

Jean-François LARUE



Jean-François LARUE né en 1963 est diplômé de l'Ecole Nationale Supérieure d'Electronique de Grenoble en 1987 et possède un DEA de Systèmes Electroniques de l'Institut National Polytechnique de Grenoble. Il a rejoint le CEA/LETI en 1989 où il a participé à la mise au point du Système SYMPATI, calculateur parallèle pour lequel a été développé un ASIC réalisé en technologie 1.2µ. Il a ensuite pris en charge le développement de nombreuses applications de vision industrielle sur ce système. Depuis 1993 il est responsable

du développement du système massivement parallèle SYMPHONIE. Ce système a nécessité le développement de deux ASICs devant être réalisés en technologie 0,5µ et intégrant chacun plusieurs centaines de milliers de portes. Un de ces ASICs est actuellement en cours de développement sous sa responsabilité.

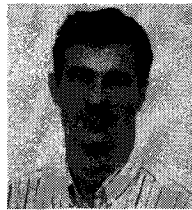
Symphonie calculateur massivement parallèle

Laurent LETELLIER



Laurent Letellier est titulaire d'une thèse de doctorat soutenue à l'INP de Toulouse en 1993, spécialité informatique. Il travaille depuis au CEA/LETI dans le Groupe Architecture Parallèle et a la charge du projet ETIC (Examen par Traitement d'Image du Combustible nucléaire). Ses domaines d'activités portent principalement sur le développement d'algorithmes d'analyse et de segmentation répondant à des besoins de contrôle non destructif par vision.

Renaud SCHMIT



Renaud Schmit né en 1957, ingénieur diplômé de l'état, il est entré au CEA en 1980 sur le centre de Saclay. Il a d'abord participé à la conception de systèmes électroniques dans le domaine de l'instrumentation nucléaire. Il a ensuite participé au développement des calculateurs de traitement d'images SYMPATI-2 et SYMPHONIE. Spécialiste du VHDL et de la conception d'ASICs, il travaille actuellement aux études sur les nouvelles architectures de processeurs.

Marc VIALA



Marc Viala né en 1965, est ingénieur diplômé du CUST en génie électrique, il possède une thèse de doctorat dans le domaine de la reconstruction 3D. Il est recruté au CEA-Leti (Direction des Technologies Avancées) en 1993. Il s'est d'abord occupé dans le cadre du projet SYMPHONIE des aspects algorithmique et logiciels associés. C'est ainsi qu'il a développé le compilateur (SPL) de ce calculateur. Depuis, il travaille sur des problèmes de reconstruction 3D et de vidéogrammétrie. Il a notamment développé une méthode originale de métrologie 3D par vision qui a fait l'objet d'un brevet.

Marc PEYTHIEUX

Marc Peythieux né en 1968, diplômé de l'Ecole Supérieure d'Electricité (SUPELEC) il effectue sa thèse de doctorat dans le groupe d'architecture parallèle du CEA/Leti sur de nouveaux concepts d'architectures multi-SIMD adaptées au traitement d'image, il a soutenu son mémoire de thèse en janvier 1998.