

# BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

N° 40 MARS 1995

SCIENCES DE L'EDUCATION ET DE L'INFORMATION

## COMITE SCIENTIFIQUE

*Patrick Abellard*  
*Jalal Almhana*  
*France Chappaz*  
*M'hamed Charifi*  
*Roger Cusin*  
*Bernard Goossens*  
*Patrick Isoardi*  
*Jean - Philippe Lehmann*  
*Nadia Mesli*  
*Patrick Sanchez*  
*Rolland Stutzmann*

## DIRECTEUR

*Edmond Bianco*

## REDACTEUR EN CHEF

*Jean - Michel Knippel*

## REDACTEUR ADJOINT

*Sami Hilala*

1 EDITORIAL,

*par Edmond Bianco*

3 OUPPI-3 : UNE MACHINE SIMD POUR LES  
MODELES DE PHYSIQUE BASES SUR LES  
AUTOMATES CELLULAIRES A 2-D,

*par Michel Cotton et Annie Perez*

5 PFLUX : UN LOGICIEL D'ANALYSE DES  
RESEAUX DE PETRI A FLUX DE DONNEES,

*par Patrick Abellard*

19 VOZZAVEDIBISAR,

*par Jean - Michel Knippel*

Publication gratuite trimestrielle de l'Université d'Aix - Marseille II  
58, boulevard Charles Livon. F - 13284 Marseille Cedex 07  
Téléphone : (33) 91 39 65 00 Télécopie : (33) 91 31 31 36

Edition 1997

ISSN 0291 - 5413

# BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

N° 40 MARS 1995

SCIENCES DE L'EDUCATION ET DE L'INFORMATION

## COMITE SCIENTIFIQUE

*Patrick Abellard*  
*Jalal Almhana*  
*France Chappaz*  
*M'hamed Charifi*  
*Roger Cusin*  
*Bernard Goossens*  
*Patrick Isoardi*  
*Jean - Philippe Lehmann*  
*Nadia Mesli*  
*Patrick Sanchez*  
*Rolland Stutzmann*

## DIRECTEUR

*Edmond Bianco*

## REDACTEUR EN CHEF

*Jean - Michel Knippel*

## REDACTEUR ADJOINT

*Sami Hilala*

**1 EDITORIAL,**

*par Edmond Bianco*

**3 OUPPI-3 : UNE MACHINE SIMD POUR LES  
MODELES DE PHYSIQUE BASES SUR LES  
AUTOMATES CELLULAIRES A 2-D,**

*par Michel Cotton et Annie Perez*

**5 PFLUX : UN LOGICIEL D'ANALYSE DES  
RESEAUX DE PETRI A FLUX DE DONNEES,**

*par Patrick Abellard*

**19 VOZZAVEDIBISAR,**

*par Jean - Michel Knippel*

Publication gratuite trimestrielle de l'Université d'Aix - Marseille II  
58, boulevard Charles Livon. F - 13284 Marseille Cedex 07  
Téléphone : (33) 91 39 65 00 Télécopie : (33) 91 31 31 36

Edition 1997

ISSN 0291 - 5413

## EDITORIAL,

### INFORMATIQUE et DESARROI.

Il faut bien constater que nous vivons l'après chute des "idéologies". Comment s'exprimer alors sur la société de ce siècle évanescant sans utiliser le langage journalistique ? avec tout ce que cela comporte de vaine, de mal digéré, de formules-choc. Comment parler avec détachement d'un système commercial où tout a tendance à devenir marchandise, où même un Président de la République, censé être un garant des lois, se transforme en voyageur de commerce au service des grands groupes commerciaux et industriels de pression ?

Comment enfin, discerner dans cette société d'abondance l'inadmissible et l'intolérable, quand l'abondance est obtenue précisément en multipliant les produits pour abaisser les coûts, si l'Homme dès l'instant qu'il revêt sa tenue de travail devient une marchandise surabondante. Obtenue selon les principes du lapinisme à la Debré, le père, l'homme à l'entonnoir sur la tête, celui qui, fidèle à ses principes a favorisé la quantité au détriment de la qualité en faisant des jumeaux.

Avec diversité et avec des bonheurs divers, le Journaliste fait son travail, qui vaut ce qu'il vaut. Entre les conflits qui ravagent certains coins privilégiés de la planète, et les annonces du premier ministre qui nous informe de quelques rançons supplémentaires, il faut bien meubler, alors on redécouvre l'importance de l'illettrisme. Quelle confiance accorder à un tel magma, d'autant qu'il n'est pas si lointain le temps où l'Académicien aux oreilles d'éléphant et aux yeux de poisson mort, alors ministre de l'information tenait la télévision à la botte de Mongénéral. Le même qui vient pavaner chez Pivot en essayant de minimiser l'importance du laobai, le goulag des chinois rouges-sang, simplement parce que la chine représente un fabuleux marché potentiel, et qu'il ne faut pas inquiéter les spectres à la Deng xiao ping.

Cohérence contre cohérence. La cohérence du système commercial qui apporte effectivement une sorte de bien-être au prix d'un gaspillage pharamineux d'hommes et de richesses, et puis en face, qui fut un instant l'espoir des peuples, un totalitarisme ravageur à base d'une population inculte, qui, sous prétexte de révolution n'a su que reproduire et amplifier ce qu'on lui avait montré jusque là sans lui permettre d'y accéder: la trique et la combine.

De là à déduire que puisque l'idéologie a porté de mauvais fruits dans ses applications, la seule bonne solution se réduit au système libéral, il pourrait y avoir une marge. Sans être un admirateur inconditionnel ni de l'EDF, ni du CEA, on peut raisonnablement se demander ce que serait le marché du nucléaire livré sans contrôle à une exploitation privée. Ce qui est en partie le cas avec la COGEMA, je fais référence à la découverte des rejets de l'usine de la Hague. Par marée exceptionnellement basse, on a découvert une tubulure terriblement radioactive traversant une plage banale. Quelques rares protestations se sont élevées sur le fait qu'aucune balise ne marque l'approche de ce danger. Je ne crois pas qu'aucun ministre, député ou sénateur ait perdu le sommeil pour autant. Mais je n'ai entendu personne poser la question de savoir qu'est ce qui coule dans ce tube et comment cela se diffuse au fond de la mer. Etrange non ?

Là dessus l'informatique surgit avec ses fantastiques moyens de transport d'information n'ayant plus aucune commune mesure avec la transmission des dépêches d'agences. En un premier temps le citoyen va pouvoir plonger dans des masses considérables de données qui lui sont servies à domicile, qui lui coûtent finalement moins cher que de s'abonner à un quotidien, et dont il peut conserver toutes les traces qu'il veut sous un emballage extraordinairement réduit. En un second temps, pour survivre, ce lui sera une obligation de s'immerger dans ce qui sera de fait son image de l'univers. Et dans un troisième temps, schizophrène parfait, il vivra dans deux mondes parallèles de plus en plus divergeants, dont le premier, le monde réel ira en dépérissant.

*Edmond Bianco*

## OUPPI-3: UNE MACHINE SIMD POUR MODELES DE PHYSIQUE BASES SUR LES AUTOMATES CELLULAIRES A 2-D.

3

M. COTTON et A. PEREZ

Laboratoire d'Electronique, Institut Charles FABRY, UNIVERSITE DE PROVENCE  
Case 241, Centre de St Jérôme, 13397 MARSEILLE cedex 13

OUPPI-3, "Ordinateur Utilisant des Processeurs Parallèles Intégrés" est un prototype de machine SIMD capable d'émuler des automates cellulaires ("AC") à deux dimensions de 128 Ksites. Il est destiné à l'étude de modèles basés sur les AC déterministes à 2-D dans le domaine de la physique statistique.

Les machines SIMD permettent de traiter des AC plus complexes que celles basées sur la simple consultation de tables ("table lookup"). Les machines "universelles" SIMD à faible granularité disponibles actuellement, telles que la Connection Machine ou le DAP représentent un investissement élevé. Aujourd'hui, pour résoudre certains problèmes, on peut construire des machines SIMD spécifiques d'un coût nettement plus faible grâce à des circuits, tel le GAPP commercialisé par NCR, qui intègre 72 processeurs élémentaires travaillant en parallèle et disposés selon un réseau à maille carrée.

Des algorithmes d'automates cellulaires ont été étudiés au sein de notre équipe pour simuler des modèles de changement de phase, en particulier dans le domaine du magnétisme. Ces simulations conduisent à des grandeurs physiques dont la précision dépend du nombre de sites du réseau d'automates cellulaires. Les physiciens considèrent qu'un réseau d'un million de sites est un minimum dans ce domaine. En affectant un processeur élémentaire à chaque site pour calculer l'état futur de l'AC, il faudrait un réseau d'au moins un million de processeurs. Bien que depuis quelques années on ait commencé à étudier l'intégration de plusieurs centaines de processeurs sur une tranche de lingot de silicium, on ne produit pas commercialement, pour l'instant, des circuits avec un aussi grand nombre de processeurs.

Nous avons considéré le cas d'un réseau de quelques dizaines de processeurs seulement, appelé bloc de calcul ("BC"), associé avec un grand réseau ("GR") de sites d'AC de l'ordre d'un million de sites. Le BC traite un sous ensemble du GR appelé "pavé". Le calcul de l'état suivant du GR se fait par traitement successif de tous les pavés du GR par le BC. La notion de voisinage implique cependant l'utilisation d'un "anneau de garde" autour de chaque pavé.

Nous avons donc entrepris la réalisation d'OUPPI-3 pour mettre en oeuvre ces idées et rendre le découpage en pavés, ainsi que la gestion de l'anneau de garde, automatiques et transparents pour l'utilisateur de la machine. Ce prototype, actuellement en cours de test, est conçu pour être évolutif:

- la mémoire de données du GR est extensible de 128 koctets à 2 Moctets,
- pour un nombre fixé de sites du GR, le nombre de lignes et de colonnes est modifiable,
- l'état d'un site du GR est codé sur 8 bits et il est extensible à 12 bits,
- le BC est extensible de 144 à 576 processeurs.

### Bibliographie succincte:

A. PEREZ, M. COTTON, G. ROGER, J. HERVE, "OUPPI-1, a SIMD Computer using integrated parallel processors" in COMPAR 88, Edited by C.R. JESSHOPE and K.D. REINARTZ, Cambridge University Press, pp 205-213 (1989).

O. PARODI, H. OTTAVI, M. COTTON, J. HERVE, Economical Parallel Processing Symposium, Berne (31 mai-1 juin 1988), "Statistical Mechanics on GAPPs".

M. COTTON, A. PEREZ, G. ROGER, O. PARODI, J. HERVE, H. OTTAVI, Congrès de la Société Française de Physique, Lyon (septembre 1989), "OUPPI-3, Calculateur Parallèle dédié aux simulations sur réseaux".



# PFLUX : UN LOGICIEL D'ANALYSE DES RESEAUX DE PETRI A FLUX DE DONNEES

P. ABELLARD. Laboratoire d'Automatique et d'Informatique Appliquées de Toulon  
Université de Toulon, BP 132, 83957 LA GARDE CEDEX.

## 1 - INTRODUCTION.

Le problème du calcul parallèle comporte plusieurs aspects différents, souvent liés qui convergent vers les mêmes buts : augmenter, sur les plans matériel et logiciel, les performances des systèmes parallèles à évolutions simultanées et faciliter leur mise en oeuvre.

Cependant, ce problème repose sur deux éléments principaux :

- La *modélisation* des programmes parallèles et de leurs structures matérielles de traitement,
- Le *choix* et la *conception* d'une architecture parallèle capable d'exécuter des opérations évoluant en même temps.

Le degré de lien entre ces deux éléments joue un rôle important dans la mise en oeuvre d'un processus parallèle car l'architecture matérielle doit *s'adapter* au modèle de représentation du calcul parallèle et non pas le contraire.

Ainsi, à l'heure actuelle, l'*architecture à flux de données* semble, de part ses fonctionnalités, parfaitement bien adaptée à la réalisation de calculs parallèles avec efficacité. Sa modélisation par *Réseaux de Petri à Flux de Données* facilite grandement, avec le logiciel PFLUX, son implémentation sur des machines de type multiprocesseur à flux de données.

## 2 - RAPPELS [ALMHANA 1983].

### 2.1 - Architecture à flux de données.

Parmi les différentes architectures possibles, nous avons retenu l'*architecture à flux de données* qui ne présente pas les inconvénients des architectures multiprocesseurs conventionnelles :

- *Concurrence limitée* : l'utilisation du modèle de Von Neuman, caractérisé par un mode de fonctionnement séquentiel et un partage des zones mémoires, réduit la concurrence entre les différentes tâches du programme et par conséquent le nombre de tâches exécutées simultanément.

- *Système de contrôle complexe* : il est nécessaire pour gérer les communications processeurs-processeurs et processeurs-mémoires et pour résoudre les conflits d'accès aux ressources. Des phénomènes d'étranglement ou de saturation de bus compliquent ce contrôle et diminuent son efficacité.

- *Difficultés de programmation* : le principal problème de la programmation parallèle dans les machines multiprocesseurs est d'éviter les interférences entre les différentes instructions du programme. Or celles-ci sont importantes dans le cas d'un programme conventionnel exécuté par un système multiprocesseur, ce qui rend l'analyse du comportement du programme et la mise en évidence de problèmes particuliers tels que le non-déterminisme des calculs, la réentrance d'opérations et l'existence de blocages, plus difficiles et plus complexes.

L'architecture à flux de données (figure 1) est mieux adaptée aux calculs concurrentiels pour les raisons suivantes :

- l'implantation du calcul parallèle est plus facile,
- toutes les opérations constituant le programme sont locales et dépendent uniquement des opérandes dont elles ont besoin,
- il n'y a pas d'interférence entre les différentes tâches exécutées simultanément car les zones mémoires ne sont jamais partagées.

L'architecture à flux de données est structurellement différente des architectures conventionnelles et elle n'implique pas de contraintes matérielles supplémentaires sur l'exécution des programmes parallèles.

Elle ne comporte pas de processeur central : il est remplacé par une tranche de modules de traitement élémentaires, unités arithmétiques et logiques, processeurs d'entrées-sortie, etc... Elle ne comporte pas non plus de mémoire RAM : celle-ci est remplacée par une tranche de modules mémoires contenant des codes opératoires, des opérandes, des adresses, etc...

Le compteur de programme et les registres de stockage n'existent pas.

Un module de gestion et de communication de paquets d'opérations (réseau d'arbitrage) permet de relier la sortie d'une tranche de modules mémoires à l'unité de traitement appropriée du système.

Chaque unité de traitement envoie un paquet de résultats vers la section de mémoires par un réseau de distribution.

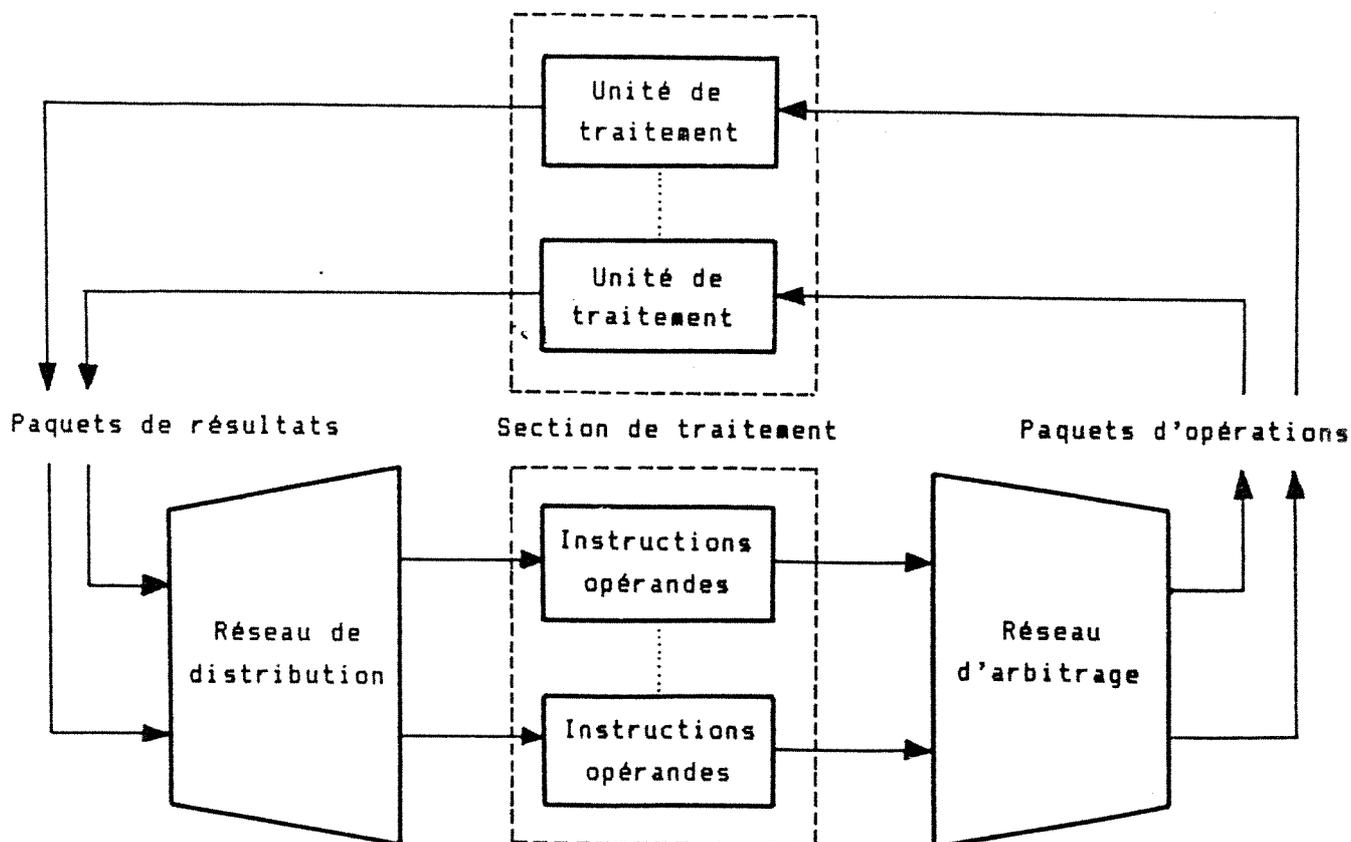


Figure 1 : Architecture à flux de données.

La réalisation d'un calcul parallèle avec efficacité sur une machine multiprocesseur nécessite une étude approfondie. Un modèle de représentation du calcul parallèle est indispensable, d'une part pour exprimer le parallélisme et d'autre part pour étudier le comportement dynamique du système matériel.

Le point important à considérer lors de la conception d'une machine parallèle, est la relation entre le modèle de représentation du programme parallèle et l'architecture à réaliser, car cette architecture doit s'adapter au modèle et non pas le modèle à l'architecture.

Parmi les modèles existants dans la littérature, nous avons choisi les Réseaux de Petri à Flux de Données qui sont parfaitement bien adaptés à la représentation de l'architecture à flux de données.

## 2.2 - Réseaux de Petri à Flux de Données.

### 2.2.1 - Description qualitative.

Dans un Réseau de Petri à Flux de Données (RdPFD), une opération est réalisée par un opérateur qui dispose d'un ensemble de données "opérandes" et qui fournit un ensemble de résultats.

Il contient deux sortes de places *variables* et *opérateurs* et se représente comme indiqué sur la figure 2 dans laquelle :

- $t_i$  et  $t_j$  sont appelées respectivement *transition d'entrée* et *transition de sortie* de l'opérateur  $O_r$  associé à la place  $p_{ij}$ .

- les places  $t_i = \{p_1, p_2, \dots, p_s\}$  représentent les données nécessaires à l'opérateur  $O_r$  et les places  $t_j = \{p'_1, p'_2, \dots, p'_r\}$  représentent les résultats qu'il fournit.

On suppose qu'une place associée à une variable ne peut avoir au plus qu'un arc d'entrée  $CARD(^{\circ}p) \leq 1$  et qu'un arc de sortie  $CARD(p^{\circ}) \leq 1$  car la possibilité de choix dans le mode d'obtention des données n'est pas admise et une donnée n'est pas explicitement partageable entre plusieurs opérateurs.

On suppose enfin que les réseaux sont *saufs*. L'arrivée d'une marque dans une place "variable" donc représentant une donnée, signifie que la valeur de la variable est créée et occupe une place en mémoire.

De même, l'arrivée d'une marque dans une place "opérateur" signifie que l'opérateur est activé et que l'opération associée est en cours d'exécution pendant un temps  $\tau$ .

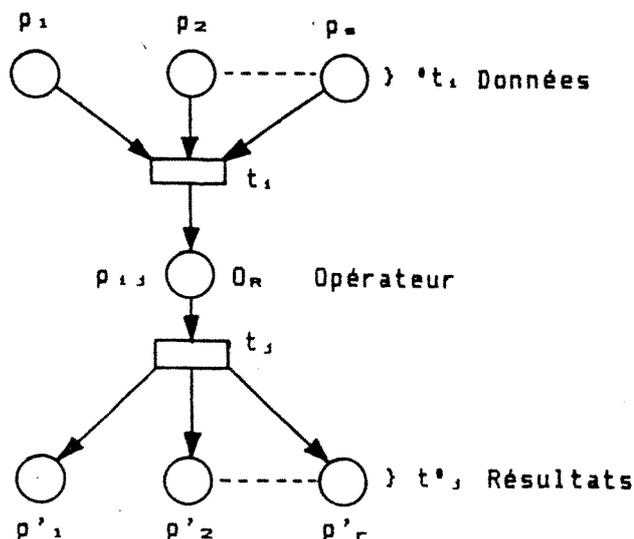


Figure 2 : Réseau de Petri à Flux de Données.

2.2.2 - Définition mathématique.

Un Réseau de Petri à Flux de Données est un septuplet  $\langle R, \varphi, \xi, \psi, X, O, C \rangle$  défini de la façon suivante :

R est un Réseau de Petri à places biparties conforme, composé d'un ensemble  $P_v$  de places variables et d'un ensemble  $P_o$  de places opérateurs, tel que :

$$\forall p \in P_v \rightarrow \text{CARD}(^{\circ}p) \leq 1 \text{ et } \text{CARD}(p^{\circ}) \leq 1$$

$$\forall p_i \in P_o, p_j \in P_o \text{ et } i \neq j \rightarrow ^{\circ}p_i \cap ^{\circ}p_j = \emptyset \text{ et } p_i^{\circ} \cap p_j^{\circ} = \emptyset.$$

$\varphi$  est une application :  $\varphi \upharpoonright P_v : P \rightarrow X$  et  $\varphi \upharpoonright P_o : P \rightarrow O$ , telle que :

$$\forall p_j \in P_o, p_i \in P_o \text{ et } \varphi(p_i) = \varphi(p_j), i \neq j \rightarrow \forall t_l \in ^{\circ}p_i, t_k \in ^{\circ}p_j$$

$$\{\varphi(^{\circ}t_l)\} \neq \{\varphi(^{\circ}t_k)\}.$$

Ainsi deux opérateurs identiques ne peuvent pas travailler sur le même ensemble de données.

$\xi$  est une application injective  $\xi : X \rightarrow \mathcal{M} = \{ME_1, ME_2, \dots, ME_u\}$  telle que :

$$\forall p \in P_v, \exists ME \in \mathcal{M} \text{ tel que } ME = \xi(\varphi(p)), \mathcal{M} \text{ est appelé ensemble de zones mémoires.}$$

$\psi$  est une application,  $\psi : T \rightarrow C$

$X = \{x_1, x_2, \dots, x_s\}$  est un ensemble de variables (réelles, entières, logiques...) prenant leurs valeurs respectivement dans les domaines  $D_1, D_2, \dots, D_u$ .

$O = \{o_1, o_2, \dots, o_v\}$  est un ensemble fini d'opérateurs définis comme des applications internes de  $D_1 * D_2 * \dots * D_u$ .

$C = \{c_1, c_2, \dots, c_r\}$  est un ensemble de prédicats sur les variables de X.

2.2.3 - Graphe des marquages.

La figure 3 montre le RdPFD du calcul  $c_{11} = a_{11} \cdot b_{11} + a_{12} \cdot b_{21}$  et la figure 4 donne le graphe des marquages associés pour l'étude du comportement dynamique du réseau.

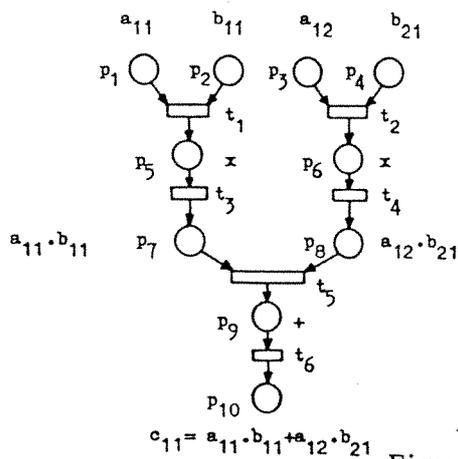


Figure 3 : RdPFD du calcul  $c_{11} = a_{11} * b_{11} + a_{12} * b_{21}$

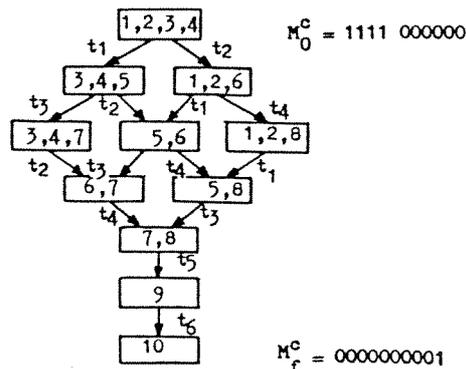


Figure 4 : Graphe des marquages associés.

L'ensemble des opérations d'édition, d'ordonnancement, de simulation... a été automatisé avec le logiciel PFLUX.

### 3 - PFLUX [ABELLARD, SALGON, RICHON, BALMAT 90].

PFLUX est un logiciel, entièrement graphique, d'analyse des Réseaux de Petri à Flux de Données. Développé en Pascal, il a été conçu pour fonctionner sur tout micro-ordinateur type compatible équipé d'une carte graphique EGA/VGA.

Le menu principal donne accès à l'ensemble des fonctions de PFLUX :

- La commande *Edition* permet d'éditer, de corriger et de visualiser un RdPFD grâce aux commandes *Edit bloc*, *Correction* et *Visualisation*. *Edit bloc* permet l'édition du réseau sous la forme de "blocs" constitués d'un opérateur, avec ses transitions d'entrée et de sortie et deux ensembles de places variables constituant les données à traiter et les résultats obtenus (figure 5). La saisie du réseau s'effectue opérateur par opérateur pour lesquels il est nécessaire de préciser le nombre de places *variables* en entrée et en sortie et d'indiquer chaque élément. La commande *Correction* permet de corriger des blocs déjà saisis pour en modifier par exemple, un indice ou une temporisation. Avec la commande *Visualisation*, le réseau en cours d'édition peut être examiné de manière *Partielle*, c'est à dire bloc par bloc, ou *Globale* pour le réseau complet.

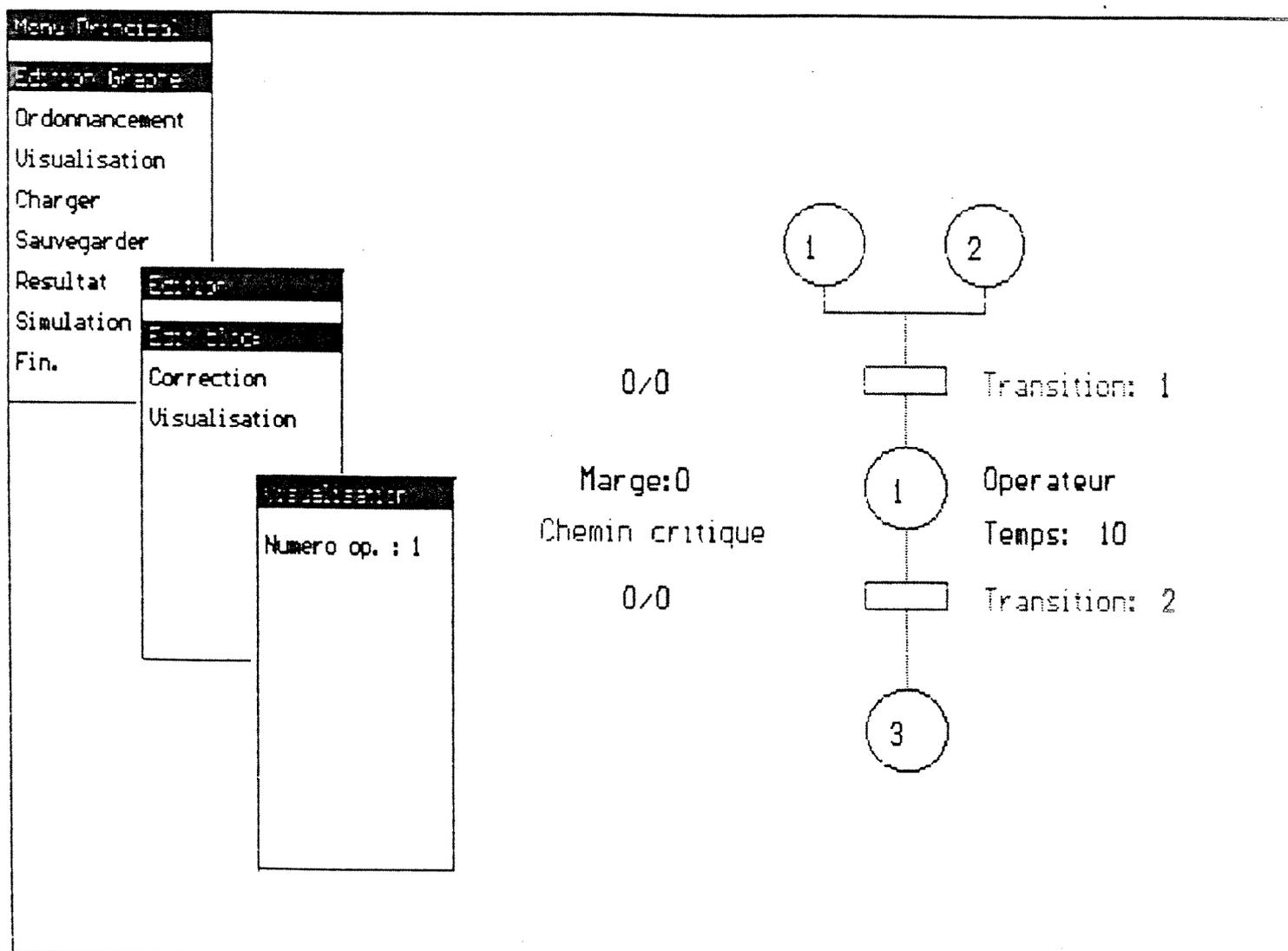


Figure 5 : Edition d'un bloc.

- Une fois que le réseau a été saisi, il est possible d'effectuer un *Ordonnancement* temporel et contrôler les résultats obtenus soit par *Visualisation globale ou partielle*, soit par *Résultats*. L'ensemble des instants de tir au plus tôt et au plus tard ainsi que les intervalles de flottement sont calculés pour mettre le chemin critique en évidence à l'*Ecran* ou sur *Imprimante*. Il est alors possible de procéder à une simulation de manière à étudier le comportement dynamique du réseau complet (figure 6).

- Tout réseau peut être sauvegardé par la commande *Archivage* et rappelé ensuite par la commande *Chargement*. Conçu pour l'étude de réseaux de Petri avant leur implémentation sur processeur à flots de données  $\mu$ PD 7281, PFLUX permet deux initialisations différentes des variables : une initialisation *Globale* (marquage correct) ou une initialisation *Séparée* caractérisant le fonctionnement réel du processeur.

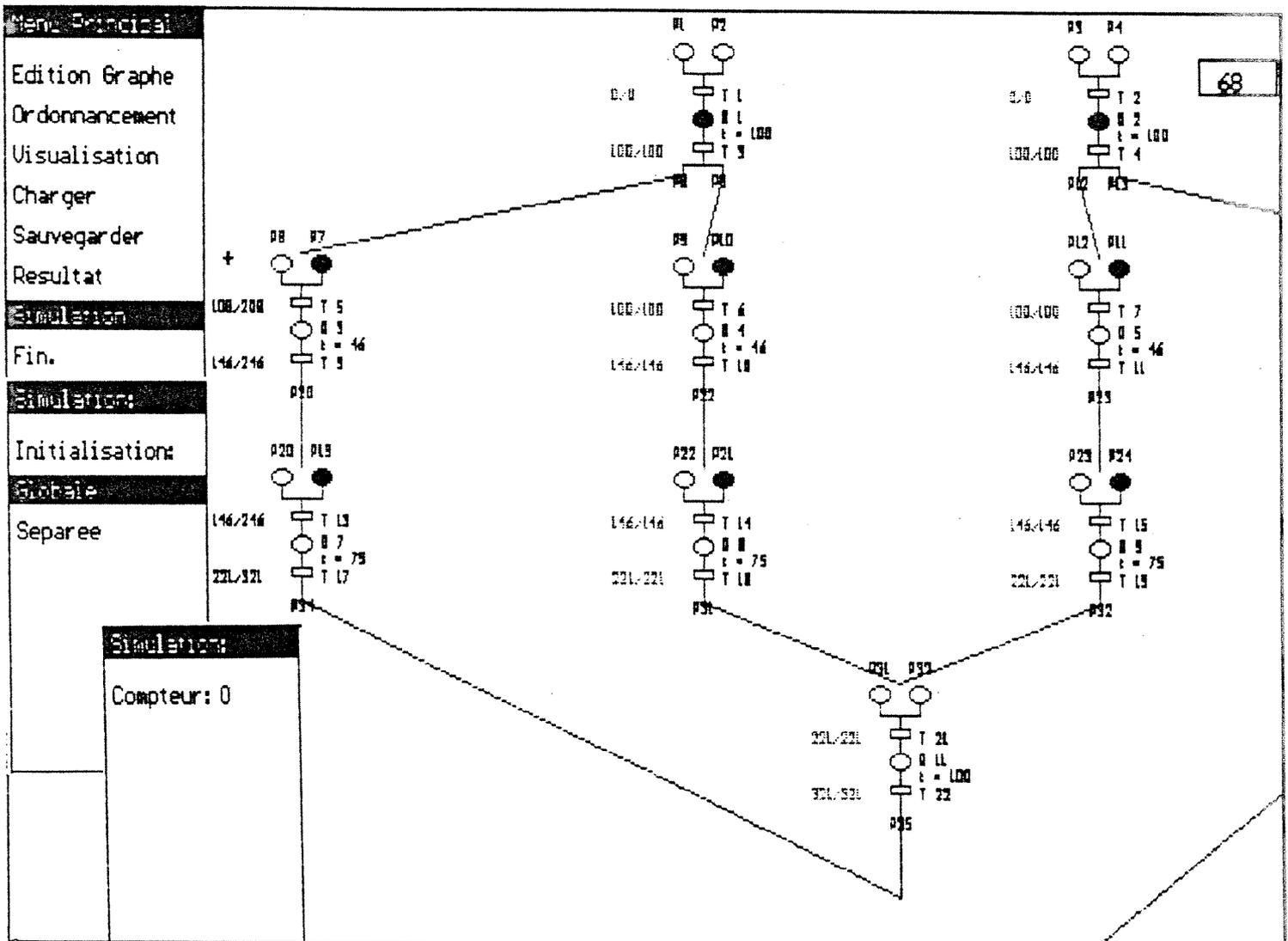


Figure 6 : Simulation d'un réseau.

Le processeur à flux de données NEC  $\mu$ PD 7281 a été retenu pour l'implémentation des algorithmes parallèles, car celui-ci reflète au niveau du composant élémentaire la macro-structure de l'architecture à flux de données précédemment définie (figure 1).

## 4 - PROCESSEUR A FLUX DE DONNEES [MESCHACH 84].

### 4.1 - Constitution.

Le processeur à flux de données ne commence sa fonction qu'après avoir reçu les données à traiter. Ces données sont combinées avec les instructions reçues après l'initialisation. Lorsque l'instruction a reçu toutes les données nécessaires pour l'exécution, elle est introduite dans une file d'attente et ensuite dans une unité de traitement.

Le résultat peut être une entrée pour une instruction supplémentaire ou une sortie si c'est un résultat final. Le **graphe de flux** de la figure 7 extrait du User's Manual 7281 de NEC, montre que les instructions sont représentées comme des noeuds, tandis que les données qui passent par les liaisons entre les noeuds s'appellent des *jetons*.

Compte tenu des blocs qu'ils traversent, les jetons ont des longueurs et des subdivisions variables. Par exemple, les jetons d'entrée et de sortie comprennent 32 bits et sont divisés en quatre zones : adresse du boitier, zone d'identification, zone de commande et zone de données (figure 8). La zone d'identification informe le processeur du calcul dans lequel le jeton doit être incorporé. La zone de commande lui indique si le jeton d'entrée contient des données à traiter, des informations de diagnostic ou s'il fait partie d'un programme objet en cours de chargement.

Après l'acceptation du jeton par le processeur (zone d'adresse du boitier correspondant au numéro du module défini après la remise à zéro), la zone d'adresse du boitier n'est plus nécessaire et les 28 bits restants forment le jeton à transmettre au bloc interne suivant. Durant le traitement, plusieurs zones sont rattachées et d'autres sont supprimées selon le type d'opération spécifiée et selon la place prise par le jeton dans le pipeline interne. Un jeton incorporé dans l'unité de traitement du processeur peut avoir une taille de 64 bits.

Le 7281 se compose de neuf blocs principaux (figure 9). Cinq d'entre eux (LT, FT, DM, Q et PU) forment un circuit pipeline dans lequel les jetons peuvent circuler et être traités plusieurs fois avant d'en sortir. La fonction de chaque bloc peut être brièvement décrite :

- *Contrôleur d'entrée (IC)* : Il commande l'entrée des jetons et décide s'ils doivent être traités dans le circuit interne ou transférés au contrôleur de sortie. Après la remise à zéro, il introduit l'adresse du boitier (numéro du module) identifiant le composant dans la chaîne. Cette adresse est comparée à la zone correspondante du jeton introduit.

- *Contrôleur de sortie (OC)* : Il transmet les jetons sous forme de deux mots de 16 bits en établissant une liaison avec le boitier suivant.

- *Table des liaisons (LT)* : Elle est composée de RAM mémorisant l'adresse des jetons suivants et l'adresse du code opératoire.

- *Table de fonctions (FT)* : Elle mémorise les paramètres des instructions/opérations.

- *Générateur d'adresses et contrôleur du flot de données (AC & FG)* : Ils créent des adresses pour la mémoire temporaire et surveillent le flot des jetons dans le processeur.

Appendix A Flowgraphs

Flowgraphs are used when creating uPD7281 programs. These correspond to the flowcharts used writing for conventional processors.

As shown in Figure A-1, flowgraphs are composed of 'arcs' and 'nodes'. Arcs indicate the flow of data to be processed by the uPD7281 and nodes indicate the type of processing to be performed with the uPD7281.

A.1 Flowgraph Explanation (outline)

Figure A-1 shows the flowgraph when a uPD7281 processor is used to execute the expression  $(x + y) \times 5$ .

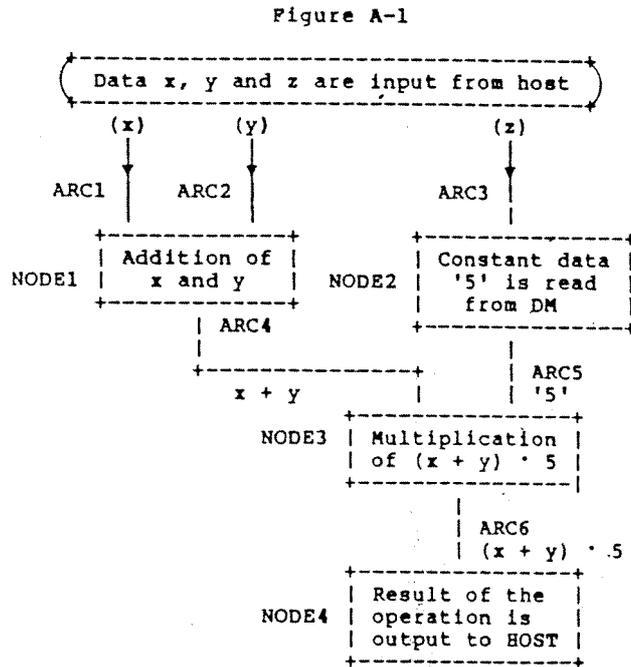


Figure 7 : Graphe de flux.

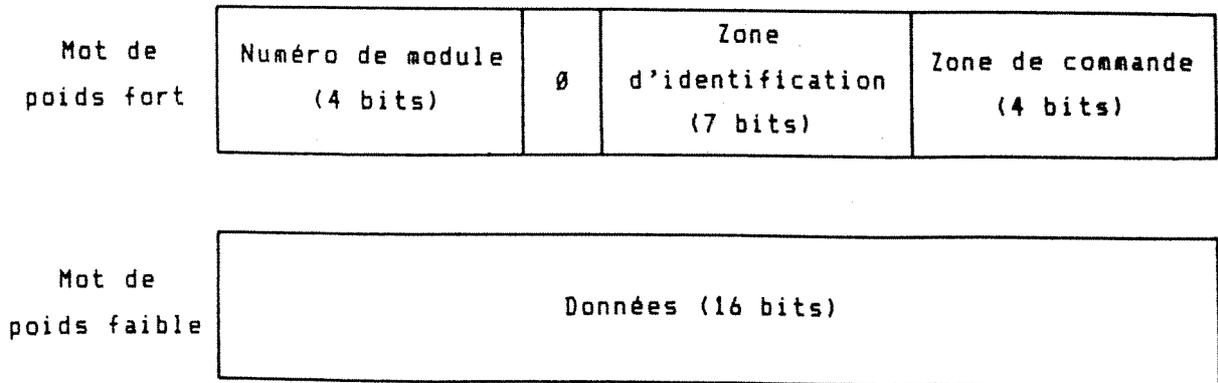


Figure 8 : Jeton d'entrée et de sortie.

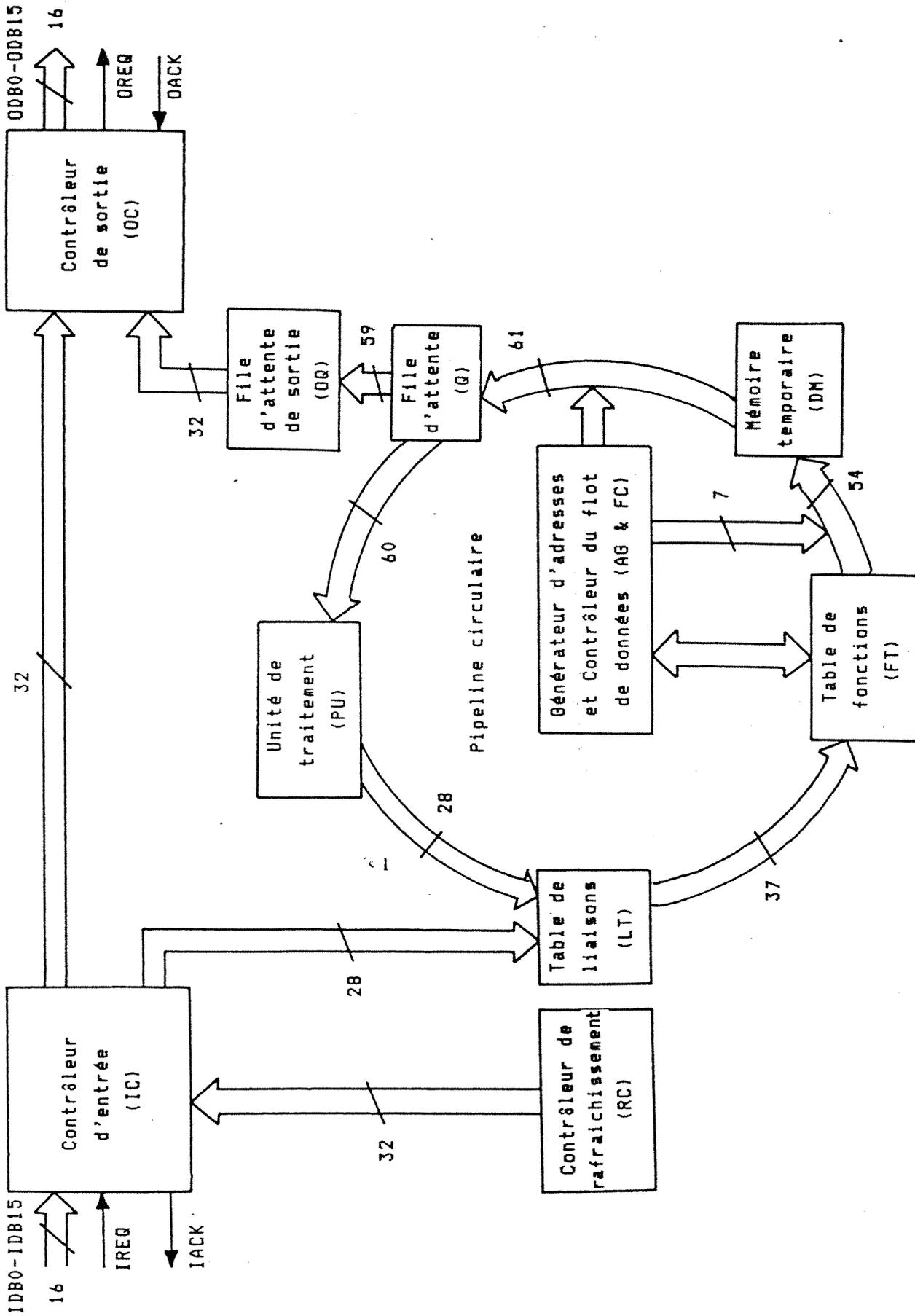


Figure 9 : Constitution interne du processeur à flux de données  $\mu$ PD 7281.

- *Mémoire temporaire (DM)* : Elle mémorise les constantes et les résultats intermédiaires.

- *File d'attente (Q)* : Elle construit une suite chronologique des jetons à traiter.

- *File d'attente de sortie (OQ)* : Elle construit également une suite chronologique des jetons, mais pour la sortie.

- *Unité de traitement (PU)* : Elle effectue des opérations logiques, arithmétiques ou des manipulations de bits d'un mot.

Les blocs sont reliés suivant une structure pipeline. Un cycle pipeline dure 200 ns et la vitesse de traitement est de 5 Mips pour un processeur. Après que le premier jeton ait été accepté, le contrôleur d'entrée décide si ce jeton passe directement au contrôleur de sortie ou s'il doit être transmis au circuit interne car le jeton contient en lui-même l'information nécessaire à cette décision. Le contrôleur de rafraîchissement RC régénère les jetons pour la mémoire RAM dynamique.

Le jeton accepté pour le traitement interne active la table de liaisons à partir de laquelle l'adresse de destination suivante est choisie. Après un cycle pipeline, le jeton atteint la table de fonctions et le code d'opération est extrait pour l'unité de traitement. Ainsi, le jeton passe dans le circuit interne et est traité dans l'unité adéquate. Il peut être traité plusieurs fois dans ce circuit avant d'en sortir. Jusqu'à sept jetons peuvent circuler simultanément dans le circuit et subir ainsi un traitement tout à fait parallèle.

Du point de vue d'un jeton, le circuit interne représente son flot personnel de traitement pipeline, ce qui signifie que ce processeur permet un traitement pipeline souple et individuel des jetons. Différents flots de traitement pipeline peuvent circuler en même temps sur le circuit interne car c'est finalement le jeton qui décide de ce qui doit lui arriver et qui dirige l'architecture du 7281.

Cette architecture à flux de données permet donc d'augmenter la vitesse de traitement et de simplifier le développement des applications pour trois raisons principales :

- a) les blocs internes fonctionnent indépendamment les uns des autres, ce qui simplifie le développement,
- b) les signaux de commande peuvent être réduits au minimum car les jetons contiennent en eux-mêmes cette information,
- c) l'indépendance des blocs permet le test individuel de chaque bloc.

#### 4.2 - les instructions.

Le jeu d'instructions peut se décomposer en quatre groupes principaux :

- *Unité de traitement (PU)* : ces instructions comprennent des opérations logiques, arithmétiques et des manipulations de bits dans la zone de données des jetons.

- *Générateur d'adresses et unité de commande du flot de données (AG & FC)* : ces instructions AG et FC exécutent des opérations de lecture/écriture dans la mémoire de données, de synchronisation des données dans celle-ci pour des opérations nécessitant deux opérands et des modifications de la zone d'identification du jeton entrant afin de contrôler le flot.

- *Instructions de sortie (OUT)* : elles sont utilisées pour extraire les jetons déjà traités dans le circuit interne.

- *Instructions de génération (GE)* : elles permettent de créer plusieurs jetons à partir d'un seul pour les introduire dans l'unité de traitement ou pour les transmettre hors du 7281.

Ces groupes d'instructions sont utilisés seuls ou combinés entre eux. Par exemple, pour une multiplication dans laquelle un opérande est le jeton d'entrée et l'autre opérande se trouve dans la mémoire de données, une instruction de l'unité de traitement (MUL : multiplication) et une instruction du générateur d'adresses et de l'unité de commande du flot de données (par exemple RDCYCS : lire la mémoire cycliquement) sont combinées. L'assemblage de ces processeurs élémentaires constitue un multiprocesseur.

#### 4.3 - Multiprocesseur à flux de données.

La structure du composant 7281 permet une réalisation simple de systèmes multiprocesseurs car deux bus de 16 bits (IDB0-IDB15 en entrée et ODB0-ODB15 en sortie) ainsi que des broches de demande d'interruption (IREQ-OREQ) et de confirmation (IACK-OACK) sont disponibles. Les autres bornes concernent l'horloge CLK, le RESET et l'alimentation.

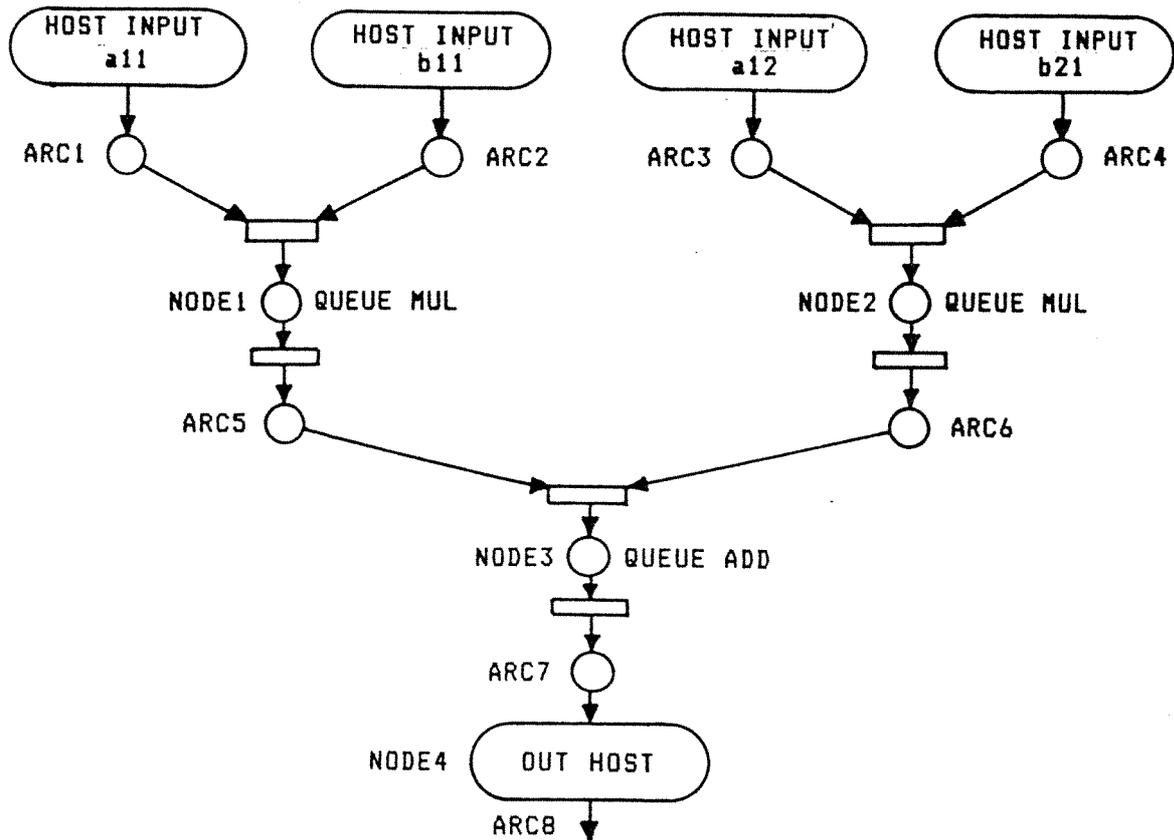
Après l'arrivée du jeton dans le premier 7281 de la cascade, la zone d'adresse du composant est comparée au numéro de module du 7281. Si ce numéro de module indique que le jeton doit être traité dans un autre 7281, le premier le transmet tout simplement au suivant. Ce processus se poursuit jusqu'à ce que le jeton atteigne le processeur destiné au traitement du jeton concerné.

Etant donné que les jetons déclenchent eux-mêmes la fonction à exécuter, il n'est pas nécessaire de s'occuper de l'état du traitement dans le 7281 suivant. Cela permet de considérer la cascade comme un module de traitement rapide dont les performances peuvent être augmentées par l'adjonction de boîtiers 7281 supplémentaires.

Les Réseaux de Petri à Flux de Données sont utilisés pour la modélisation du fonctionnement du multiprocesseur à flux de données [ABELLARD, NOLIBE, RAZAFINDRAKOTO 1990]. La figure 10 montre comment ils peuvent remplacer avantageusement pour leur validation, simulation, optimisation, les graphes de flux : les noeuds correspondent aux places opérateurs, tandis que la notion d'arc est prise au sens large de liaison inter-opérateurs par une place variable. Dans le langage fonctionnel d'assemblage utilisé, le programme source décrit la fonction à exécuter et ne spécifie pas toutes les étapes comme pour les assembleurs habituels.

## 5 - APPLICATIONS.

Parmi les différentes applications en cours de développement au laboratoire LAIAT, citons deux exemples d'application d'imagerie numérique pour le Service de Rééducation Fonctionnelle du CHU Timone de Marseille.



```

1 . EQUATE   HOST=0;
2 . MODULE   EXONE=1;
3 . INPUT    ARC1,ARC2,ARC3,ARC4;
4 . OUTPUT   ARC8;
5 . LINK     ARC5=NODE1(ARC1,ARC2);
6 . LINK     ARC6=NODE2(ARC3,ARC4);
7 . LINK     ARC7=NODE3(ARC5,ARC6);
8 . LINK     ARC8=NODE4(ARC7,  );
9 . FUNCTION NODE1=MUL,QUEUE(QUE1,1);
10. FUNCTION NODE2=MUL,QUEUE(QUE2,1);
11. FUNCTION NODE3=ADD,QUEUE(QUE3,1);
12. FUNCTION NODE4=OUT1(HOST,0);
13. MEMORY   QUE1=AREA(1);
14. MEMORY   QUE2=AREA(1);
15. MEMORY   QUE3=AREA(1);
16. START;
17. DATA    EXEC(EXONE,ARC1);
18. DATA    EXEC(EXONE,ARC2);
19. DATA    EXEC(EXONE,ARC3);
20. DATA    EXEC(EXONE,ARC4);
21. END.

```

Figure 10 : RdPFD et programme d'assemblage pour processeur à flux de données  $\mu$ PD 7281 du calcul  $c_{11} = a_{11} * b_{11} + a_{12} * b_{21}$

La première consiste en une Conception et Fabrication Assistées par Ordinateur de corsets de scoliose [ABELLARD, CONIL, DELARQUE, DUPLAIX 91]. A partir de deux images, contenant six points de référence, fournies par un système stéréoscopique, on reconstitue par des opérations de géométrie de surface le volume tridimensionnel du tronc d'un scoliotique en vue de quantifier l'évolution de sa maladie, de concevoir et de fabriquer son corset de correction (photo 1). L'architecture à flux de données a été retenue ici pour la grande quantité de calculs à réaliser.

La seconde consiste en une mesure de variations angulaires de différentes articulations, particulièrement intéressante pour celles à plusieurs degrés de liberté, puisqu'aucun capteur n'est posé sur le patient qui place simplement son articulation à rééduquer devant une ou deux caméras (photo 2). Dans cette application, l'architecture à flux de données est particulièrement efficace pour l'aspect traitement en temps réel [ABELLARD, BENEZET, VITON 91].

## 6 - CONCLUSION.

L'architecture à flux de données modélisée par Réseaux de Petri à Flux de Données offre des perspectives intéressantes en robotique, téléopération, traitement d'images, filtrage numérique, etc... partout où des calculs longs et volumineux doivent être traités rapidement sur des systèmes performants et peu coûteux pour envisager des réalisations industrielles.

MOTS CLES : Réseaux de Petri, Flux de Données, Multiprocesseurs, Calcul parallèle.

## BIBLIOGRAPHIE.

ALMHANA J : *Modélisation par Réseaux de Petri à Flux de Données. Application à la synthèse de l'opérateur de Riccati rapide.* Thèse Université Aix-Marseille III, Juin 1983.

ABELLARD P, SALGON E, RICHON C, BALMAT J.F : *PFLUX, a software tool for Data Flow Petri Net design.* The Petri Net Newsletter, n°36, pp 7-13, August 1990.

ABELLARD P, NOLIBE G, RAZAFINDRAKOTO N : *A data flow architecture for parallel computation.* International Conference PARBASE'90, Miami, IEEE Computer Press, March 1990.

ABELLARD P, CONIL J.L, DELARQUE A, DUPLAIX J : *Apport de la CAO dans la fabrication des corsets de scoliose.* Congrès National de l'Association Française d'Appareillage, Avril 1990, Hyères.

ABELLARD P, BENEZET P, VITON J.M : *Application de l'imagerie numérique à la mesure de variations angulaires de diverses articulations.* Congrès National de l'Association Française d'Appareillage, Avril 1990, Hyères.

MESCHACH M : *Data flow IC makes short work of though processing chores.* Electronic Design, pp 191-206, May 1984.

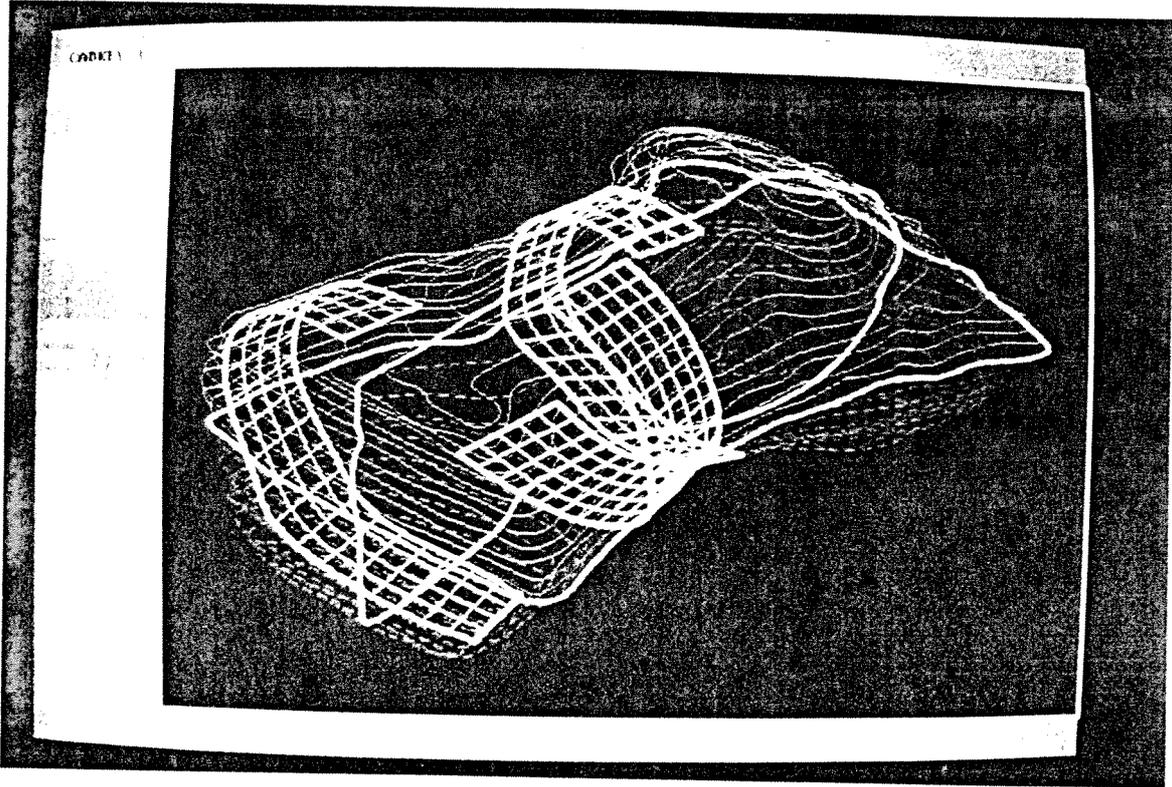


Photo 1 : CFAO de corsets de scoliose.

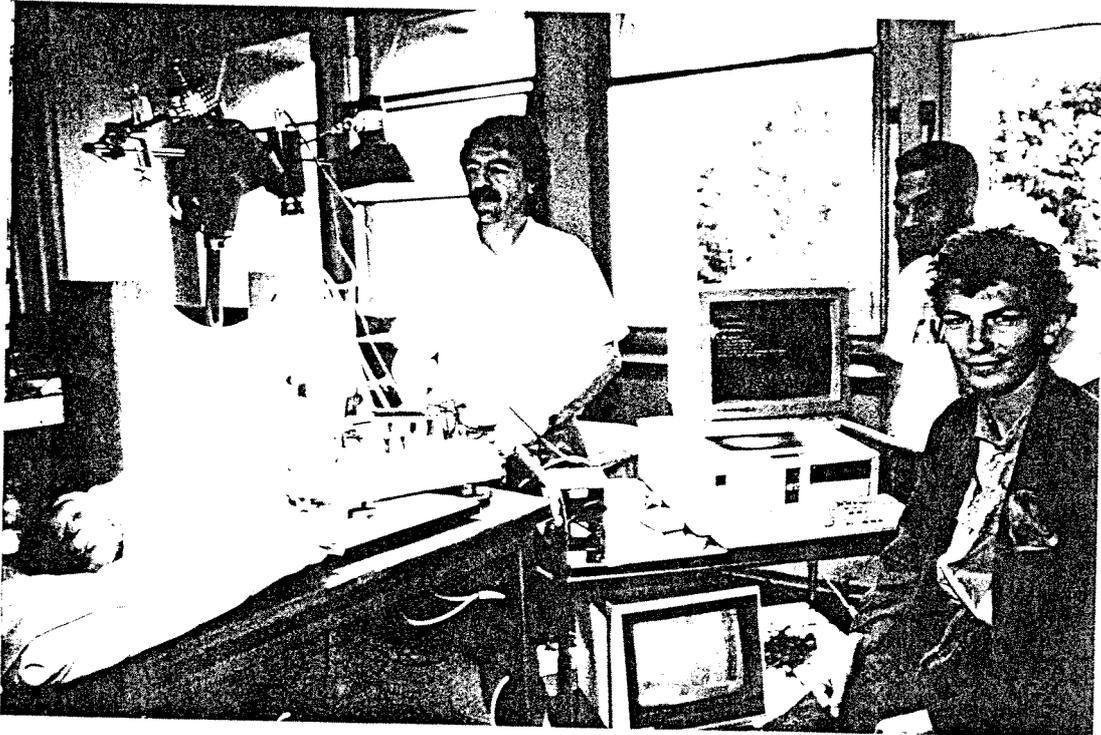


Photo 2 : Mesure de variations angulaires de pieds-bots de nourrissons au CHU Timone de Marseille.

VOUZZAVEDIBISAR,

# COMPUTATION ' 93

## JOURNEES DE TRAVAIL DU GROUPE DE FORMATION DOCTORALE

MARSEILLE 21 - 25 JUIN 1993

Organisé par

LE GROUPE DE FORMATION DOCTORALE D'INFORMATIQUE FONDAMENTALE  
ET SCIENCES DE LA COMPUTATION

L'INSTITUT D'INFORMATIQUE DE L'UNIVERSITE D'ANNABA

LE LABORATOIRE D'INFORMATIQUE THEORIQUE ET APPLICATIONS DE MARSEILLE

LE CFCIM

(CENTRE DE FORMATION ET DE CONSEIL EN INFORMATIQUE ET MATHEMATIQUES)

Sous le patronage de  
la CMEP (Comité Mixte d'Evaluation et de Prospective)

Université d'Aix-Marseille II  
Faculté Pierre Puget - LITAM  
Marseille - France

Universite d'Annaba  
Institut d'Informatique  
Annaba - Algérie

Faculté Pierre Puget. 14 Rue Puvis de Chavannes. 13001 Marseille

Téléphone : (+33) 91 13 96 29

Télécopie : (+33) 91 90 58 29

## OBJECTIF GENERAL

Ces journées de travail du groupe de formation doctorale permettent de faire le point annuellement sur des thèmes généraux de réflexion. Le courrier du CNRS, dans son dossier scientifique N°80 du mois de février 1993, présente la recherche en informatique. Par le choix des mêmes thèmes, nous permettons des rencontres entre l'université et l'industrie

### LUNDI 21/06/1993

09h30: E.BIANCO, R.CUSIN, R.DUPOUY, J.PH.LEHMANN, J.M.KNIPPEL,  
L.NOLIN, R.STUTZMANN,  
Inauguration des journées

THEME 1: LES NOUVEAUX ORDINATEURS?

Présidents : P.ISOARDI, S.GHANEMI

10h-12h: Une réflexion par B.GOOSSENS

*Déjeuner*

14h30-17h: Table ronde avec la participation des sociétés:  
ALIAS, présentation des machines ATARI  
GEMPLUS-RD2P, carte à microprocesseurs et  
objets nomades

### MARDI 22/06/1993

THEME 2: PROGRAMMER AUJOURD'HUI ET DEMAIN

Présidents : M.BADRI, J.M.KNIPPEL

10h-12h: Une réflexion par E.BIANCO

*Déjeuner*

14h30-17h: Table ronde avec la participation de l' IAO-CNRS-IMT,  
VHDL

MERCREDI 23/06/1993

THEME 3: INFORMATIQUE THEORIQUE ET MATHEMATIQUES

Présidents: P.CARTIGNY, M.T.LASKRI

10h-12h: Une réflexion par R.CUSIN

*Déjeuner*14h30-17h: Table ronde avec la participation de la société CFCIM,  
conception d'un logiciel d'aide à la décision : COMPENE II20h : *Dîner au Café parisien, place Sadi Carnot*JEUDI 24/06/1993

THEME 4: INTELLIGENCE ARTIFICIELLE

Présidents: M.SELLAMI, Y.GIRARD

10h-12h: Une réflexion par P.LIVET

*Déjeuner*14h30-16h: Table ronde avec la participation possible de l' EDF-DER,  
bases de connaissances et thésaurus

16h30-18h: Séminaire du GFDIFSC par J.M.KNIPPEL

VENREDI 25/06/1993

THEME 5: LA MACHINE COMMUNICANTE

Présidents: A.ARAAR, F.ADREIT

10h-12h: Une réflexion par R.JACQUIER

*Déjeuner*14h30-17h: Table ronde avec la participation de l'IMAG-TIMC,  
thésaurus et matière médicale homéopathique17h30-18h: Perspectives et clôture des journées  
T.BEN SBA, MME.BADRI, E.BIANCO, R.CUSIN, A.HENNI,  
J.M.KNIPPEL

**Présidence des journées**

E. BIANCO	Professeur, directeur du LITAM
D. BODIOT	Professeur, président de la CMEP
R. CUSIN	Professeur, responsable du GFDIFSC
S. HILALA	Docteur en sciences, président du CFCIM
M.T. LASKRI	Docteur en sciences, vice recteur (Annaba)

**Comité de Programme**

R. DUPUY	Professeur, Université de Paris VI
B. GOOSSENS	Professeur, Université de Paris VII
P. LIVET	Professeur, Université de Provence
L. NOLIN	Professeur, Université de Paris VII
J.B. LESSOURD	Directeur de recherches, CNRS-Marseille

**Comité d'Organisation**

E. ANDRIAMASINORO	Université d'Aix-Marseille II
E. BOUDIBA	Université d'Aix-Marseille II
S. HILALA	Université d'Aix-Marseille II
C. JOELSON	Université d'Aix-Marseille II
A. KACIMI	Université d'Aix-Marseille II
J. M. KNIPPEL	Université d'Aix-Marseille II
P. SANCHEZ	CNRS-Marseille

**Secrétariat des Journées**

E. ANDRIAMASINORO & C. JOELSON  
 Université d'Aix-Marseille II  
 LITAM-Faculté Pierre Puget  
 14, Rue Puvis de Chavannes  
 13001 MARSEILLE-FRANCE

Tél: (+33) 91 13 96 29- Fax: (+33) 91 90 58 29

Email : KNIPPEL at FRMRS11.BITNET

**Hébergement**

- Cité universitaire " Gaston Berger " , 13003 Marseille
- Résidence Sylvabelle 109, rue Sylvabelle 13006 MARSEILLE  
 ( renseignements et réservation auprès du secrétariat des journées )

**Lieu des journées**

Faculté Pierre Puget  
 GFDIFSC-LITAM  
 14, Rue Puvis de Chavannes  
 13001 MARSEILLE

SEE YOU AT THE WORK SHOP

**Université de Provence  
Atelier de Reprographie  
Centre Saint Charles  
3, place Victor Hugo  
F - 13331 Marseille Cedex 3**