AVANTAGES APPORTES PAR UNE ARCHITECTURE A FLOTS DE DONNEES DANS UN SYSTEME DE VISION ARTIFICIELLE

P. MARTIN - R. LECORDIER - L. PIEDFORT
I. GUIGUENO

I.T.E.P.E.A L.A.C.I.S , U.F.R Sciences et Techniques
L.C.I.A I.N.S.A de ROUEN Place E.Blondel BP 8 76131 Mt St AIGNAN (FRANCE)

RESUME

Nous vous proposons de présenter un opérateur programmable capable d'analyser, à la cadence de l'acquisition, les informations issues d'une caméra linéaire 2Mhz. Il est construit autour d'une configuration de huit processeurs NEC uPD7281 à flots de données reliés entre eux par un bus de données en anneau. Le uPD7281 utilise en association une architecture à flots de données et une configuration pipeline composée de modules programmables. Le concept de flots de données permet l'exécution simultanée de plusieurs traitements avec un haut niveau de parallèlisme. Ses applications sont orientées vers le contrôle de qualité et les tâches d'inspection. Cet opérateur est certes moins performant du point de vue vitesse que les systèmes câblés ou microprogrammés mais par contre il présente l'intérêt d'être flexible, de remplir des tâches plus complexes en ayant un coût moins élevé et des temps de calculs suffisants dans la plupart des cas. En dernier lieu, nous présentons une application portant sur le contrôle de 70000 gélules pharmaceutiques par heure. Cette étude permet de mettre en évidence les différents points liés à la programmation ainsi que les performances de l'opérateur.

SUMMARY

Our aim here is to describe a programmable device able of data processing from a linear 2Mhz camera, according to acquisition rhythm. It is built around a configuration of eight NEC uPD7281 processors with a data flow architecture connected together thanks to a ring data bus. The uPD7281 uses, in combination, a data flow architecture and a pipeline outline composed of programmable modules. The data flow architecture involves the simultaneous performances of several processing with a high level of parallelism. Its uses are directed towards quality control and checking tasks. However, this operator is less efficient as far as speed is concerned than wired or microprogrammed systems are but in the other hand, it has on advantage, flexibility and the ability to perform tasks which are more complex, having a cheaper cost and sufficient computation times in most cases. Lastly, we describe an application concerning the checking of 70,000 pharmaceutical capsules per hour. This study gives prominence to the different points linked to programming just as performances of this device.

1. INTRODUCTION

Le concept de flots de données a été développé pour répondre aux besoins liés aux applications portant sur un grand nombre de données telle que le traitement d'image. Le principe n'est pas nouveau puisque le MIT s'y est intéressé dès le début des années 1970 [1][2][3], on peut signaler aussi le Centre d'Etudes et de Recherche de Toulouse[4][5]. La mise en oeuvre de ce concept a été possible d'une façon aisée grâce à l'introduction sur le marché par NEC du processeur uPD7281[6][7][8][9]. Notre laboratoire a été amené à choisir ce processeur, pour compléter une chaîne d'acquisition d'image. Ces travaux se sont déroulés dans le cadre d'une étude d'inspection automatique de gélules pharmaceutiques. Cette étude a abouti à la réalisation du système de vision STIL (Système de Traitement d'Image Linéaire).

Dans cet article, nous allons dans un premier temps décrire l'architecture générale de STIL, ensuite nous présenterons l'unité de traitement temps réel et nous terminerons par l'application sur les gélules.

2. DESCRIPTION DU SYSTEME DE VISION STIL

L'opérateur image (fig. 1) est structuré autour des divers sous-ensembles suivants:

- un séquenceur microprogrammé
- un plan mémoire acquisition
- un plan mémoire rafraîchissement écran
- une unité de traitement
- une interface au standard VME

Le séquenceur génère les signaux de commandes nécessaires à une caméra linéaire CCD 1728 pixels à sortie numérisée sur 6 bits. D'autre part, il gère la mémorisation des lignes dans le plan mémoire image et fait fonction de générateur d'adresses.

La mémoire image est composée de vidéo RAM[10]. Ces mémoires sont constituées d'un registre à décalage et d'une matrice autorisant ainsi deux types d'accès. Cet avantage est mis à profit pour mémoriser la ligne en cours d'acquisition par un accès séquentiel suivi d'un transfert du contenu du registre à décalage vers la matrice. Il permet au séquenceur de gérer l'ordonnancement des lignes déjà acquises en faisant des décalages par bloc de 512 pixels. L'unité de traitement accède à cette mémoire en mode aléatoire par mot de 16 bits de façon à réduire le nombre d'accès. Ce plan mémoire est la source commune de données aux huits processeurs de l'unité de traitement.

La mémoire rafraîchissement écran, également réalisée en VRAM, assure la visualisation de l'image enregistrée par la caméra avant ou après traitement avec une définition de 512 x 512 points. Cette partie n'est pas indispensable au fonctionnement de l'ensemble, son rôle est de contrôler le défilement sous la caméra et d'aider à effectuer les réglages relatifs à l'éclairage et à l'objectif. Les données proviennent de l'unité de traitement et la reconstruction de l'image est assurée par un contrôleur

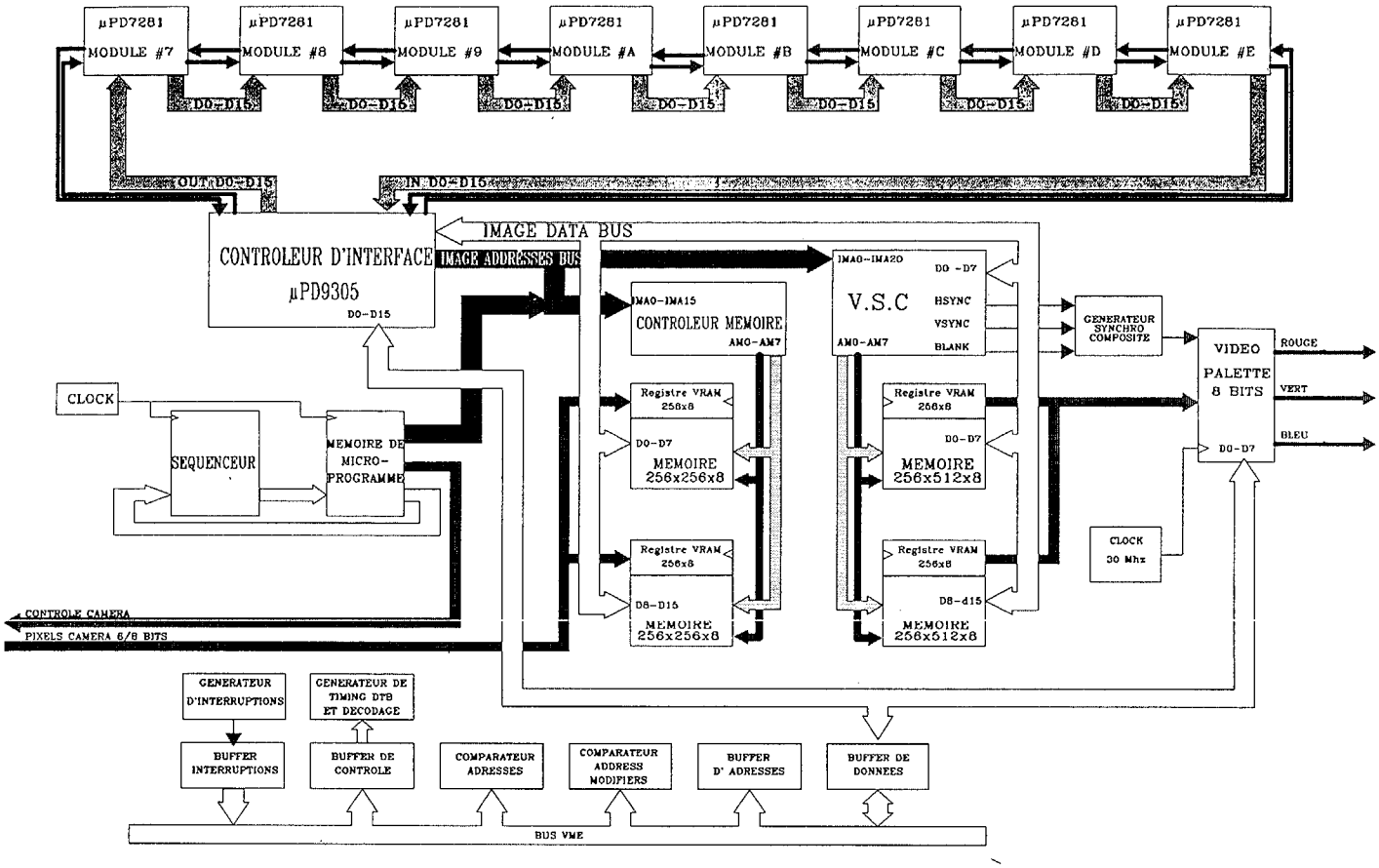

FIGURE 1

de mémoire spécialisé[11].

L'unité de traitement est constituée de huit processeurs à flots de données uPD7281 qui communiquent entre eux par l'intermédiaire d'un bus en anneau unidirectionnel dans lequel circulent les données sous la forme de jetons. L'ensemble peut fournir une puissance de calcul équivalente à 40 MIPS soit 5 MIPS par processeur. Un circuit complémentaire, nommé uPD9305 ou MAGIC (Memory Access and General bus-Interface Chip), est inséré dans l'anneau pour assurer la gestion des communications entre les processeurs et les deux plans mémoires ainsi que celles du bus VME. Cette partie sera étudiée plus en détails dans le paragraphe suivant.

Vu du bus VME par le processeur hôte, STIL se présente comme un module esclave, l'unité centrale est disponible la plupart du temps. Le CPU hôte, dans notre cas un MC 68000, n'est sollicité que lors de la phase d'initialisation pour charger les programmes exécutables dans les uPD7281. Les seuls autres échanges se font par interruptions soit pour signaler la fin d'acquisition d'une ligne soit lorsqu'il y a une anomalie détectée par les processeurs à flots de données.

3. DESCRIPTION DES PROCESSEURS [12][13]

Pour exécuter une instruction avec un processeur classique de type von Neumann, le code opération doit d'abord être lu en mémoire. Ensuite il doit être décodé et la donnée recherchée pour aboutir enfin à l'exécution.

Par contre les processeurs à flots de données exécutent, après chargement du programme objet, les instructions dès que les données nécessaires se présentent devant elles. En fait les données qui circulent à l'extérieur du uPD7281 se composent de deux mots de 16 bits, le premier mot comprend des identificateurs (numéro de module et une adresse interne à ce module) et le deuxième, la donnée elle-même. L'ensemble s'appelle jeton. Les jetons circulent sur l'anneau jusqu'à ce qu'ils trouvent leur destination qui peut être le circuit MAGIC ou bien l'un des huit modules.

La structure interne du uPD7281 (fig. 2) se décompose en trois parties :

- les contrôleurs d'entrées sorties
- l'unité de calcul
- le contrôleur de flots de données

Ce dernier est composé de cinq blocs où on peut suivre le parcours d'un jeton à l'intérieur même du processeur lorsqu'il a été accepté par le contrôleur d'entrée:

- la table de lien (LT) : l'identificateur ID du jeton entrant correspond à une adresse de cette table. A cette adresse le jeton est modifié par un nouvel identificateur ID' qui déterminera sa prochaîne destination dans la table après un tour de pipeline. De plus il reçoit une adresse code opération qui lui précise sa destination dans le bloc suivant.

- la table de fonction (FT) : à ce niveau, le jeton reçoit un nouveau champ contenant le code opération de l'instruction qui sera exécutée au bloc PU.

- la mémoire de données (DM) : ajoute une deuxième donnée au jeton si l'instruction porte sur deux opérandes. Elle peut aussi servir de mémoire locale.

- la file d'attente (Q) : contrôle l'entrée des jetons dans l'unité de calcul.

- le générateur d'adresses et contrôleur de flots (AG & FC) : fournit les adresses de DM et met à jour les champs temporaires des jetons.

Après avoir traversé l'unité de calcul, le jeton arrive à nouveau au bloc LT mais cette fois avec l'identificateur ID' pour effectuer un nouveau tour de pipeline, être détruit ou être envoyé vers le contrôleur de sortie. Durant ce trajet, le jeton change de format entre chaque bloc. Si tous les blocs sont actifs à la fois, l'unité de calcul fait une opération à chaque cycle de pipeline et réduit le temps apparent de traitement d'une donnée.

Le chargement du programme objet consiste à initialiser les tables DM, FT et LT.

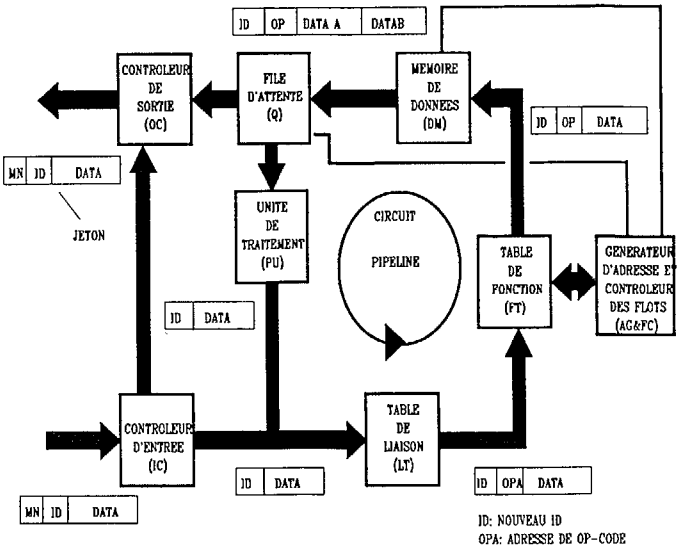

FIGURE 2

4. PROGRAMMATION DES PROCESSEURS[14]

Les programmes sont décrits par des graphes de flots de données (fig. 3). Ces graphes sont faits de noeuds, qui représentent les opérations, reliés entre eux par des arcs indiquant le cheminement des données. En réalité, les arcs représentent les identificateurs ID adressant la table de liens et les noeuds représentent les entrées dans la table FT. La principale difficulté, lors de la programmation, est certainement l'établissement du graphe de flots de données. On doit non seulement respecter l'algorithmique, mais il faut aussi synchroniser certaines instructions avec l'arrivée des jetons car on ne connait pas à priori l'ordre exact dans lequel les instructions seront effectuées contrairement à un processeur de type séquentiel.

L'exemple présenté en figure 3 est un cas typique du problème soulevé précédemment. Ce programme (fig. 3a) doit compter le nombre de jetons entrant par l'arc A0. Pour cela les jetons de l'arc A0 sont dupliqués par le noeud N1 en A1 et A2. Les copies des jetons A0 en A2 provoquent le comptage. Le noeud N2 lit l'adresse COMPTEUR dans DM et incrémente sa valeur de 1 puis le nceud N3 place cette nouvelle valeur à la même adresse. Ce programme ne semble pas poser de problème particulier et pourtant son déroulement diffère suivant la façon dont les jetons arrivent en arc A0.

Supposons que deux jetons successifs arrivent par l'arc A2,

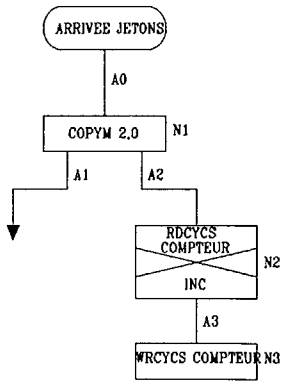

FIGURE 3a

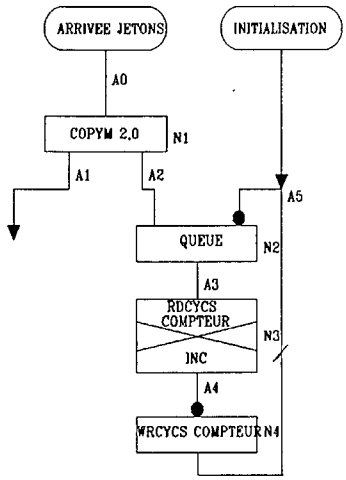

FIGURE 3b

l'emplacement COMPTEUR est alors lu deux fois consécutivement sans avoir été incrémenté ce qui donnera un résultat erroné. Pour éviter ceci, on doit ajouter une instruction supplémentaire (fig. 3b) représentée par le noeud N2 qui est une file d'attente de type FIFO dont la profondeur est limitée à 16 niveaux. Ce noeud autorise l'apparition d'un jeton en arc A3 seulement quand une donnée est présente en A2 et une autre en A5, ainsi COMPTEUR ne peut être lu qu'après avoir été incrémenté.

Toutefois si le nombre de jetons présents sur l'arc A2 croît trop rapidement, on risque un débordement de la file d'attente, dans ce cas il faut synchroniser le programme en amont ou restreindre l'arrivée des données dans les processeurs. Un simulateur est livré par NEC pour mettre au point les programmes, il permet en particulier de visualiser la circulation des jetons à l'intérieur du processeur et d'évaluer les temps d'exécutions.

Cet exemple montre aussi la difficulté que pose l'élaboration d'un langage évolué tel que C ou PASCAL qui devra en plus optimiser le partage des tâches entre les processeurs. Actuellement un assembleur graphique est en cours de développement dans notre laboratoire, il traduira les graphes de flots de données en language assembleur uPD7281.

5. APPLICATION AU CONTROLE DE GELULES PHARMACEUTIQUES[15]

Cette application nous a permis de valider l'opérateur STIL. Les algorithmes seront décrits brièvement. Nous insisterons plus particulièrement sur l'influence de la répartition des tâches entre processeurs, des accès mémoires et de l'encombrement du bus sur les performances du système.

5.1 PRESENTATION DE L'ETUDE

Une gélule est constituée de deux parties assemblées lors du conditionnement du produit pharmaceutique, elles se nomment la coiffe et le corps (fig. 4). Les principaux défauts à reconnaître sont :

- les gélules déformées
- les demi-gélules : une coiffe ou un corps seul
- les gélules vides
- une mauvaise couleur de la coiffe, défaut très important car significatif d'une erreur de produit pharmaceutique

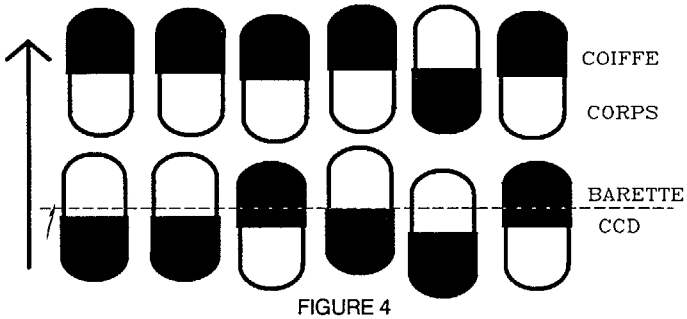

FIGURE 4

La cadence d'inspection est fixée a 70.000 gélules par heure. Les gélules se présentent sur six voies paralèlles sous la caméra et sont analysées à raison de trois par voie toutes les secondes.

5.2 REPARTITION DES TACHES ENTRE LES PROCESSEURS

5.2.1 LE PRETRAITEMENT

Le premier processeur de l'anneau effectue un prétraitement qui consiste à regrouper les pixels de la ligne issue de la caméra quatre par quatre en faisant une moyenne. Cela à l'avantage d'améliorer le rapport signal/bruit et de réduire la dimension de l'image.

Ce programme se décompose en trois parties essentielles :
- la génération des adresses sources pour la lecture de la mémoire image
- la génération des adresses destinations préparant les adresses pour la sauvegarde des données traitées
- la partie calcul effectuant le traitement lui même.

Ces trois parties s'exécutent concurremment, mais leur évolution est différente. Il faut donc les synchroniser entre elles afin d'éviter l'accumulation de jetons dans les files d'attentes. Les données sont lues

par paquet de seize mots, lorsque la dernière donnée est en cours de traitement, seize nouvelles adresses sont générées. Les files d'attente sont ainsi toujours alimentées en données, le processeur ne les attend pas si le temps de calcul de l'algorithme est supérieur aux temps d'accès mémoire.

Dans notre cas, le calcul de la moyenne d'une ligne nécessite 2,52 ms en utilisant un seul processeur. Une deuxième approche du problème consiste à charger plusieurs processeurs avec le même programme et à partager la ligne vidéo à traiter entre eux. Le temps d'exécution tombe alors à 1,26 ms en utilisant deux processeurs et à 0,92 ms avec quatre. On peut remarquer que l'utilisation de deux processeurs divise le temps d'exécution de moitié alors que celle de quatre apporte un facteur de gain de temps à peine égal à trois. Ceci s'explique par le fait que les accès mémoire sont de nature séquentielle et par conséquent le temps global d'accès pendant le traitement d'une ligne vidéo n'est pas compressible. Nous l'avons évalué à 0,71 ms. Ainsi le temps calcul de la moyenne ne pourra pas être inférieur à cette valeur.

Le simulateur nous a permis de faire une série de mesures pour évaluer l'influence du temps de cycle des mémoires sur le temps total d'exécution du calcul de la moyenne. Si on simule l'algorithme précédent avec des mémoires ayant un temps cycle nul, les temps d'exécutions sont alors de 2,42ms, 1,22ms et 0,65ms suivant le nombre de processeurs employés. Les deux premiers temps sont très proches de ceux mesurés sur le prototype et montrent que l'architecture utilisée est peu sensible au temps de cycle des mémoires alors que le troisième temps est très inférieur à celui mesuré. Dans cette dernière situation l'architecture à flots de données rejoint l'architecture plus classique de type von Neumann puisque les processeurs sont obligés d'attendre les données. Pour y remédier, il serait préférable que les données se situent dans la mémoire interne du processeur.

L'activité des processeurs est déterminée par le taux d'occupation de leur bus interne. Ce taux est de 69% quand on utilise un ou deux processeurs alors qu'il descend à 48% avec quatre processeurs. C'est la conséquence de l'encombrement au niveau des accès mémoire. Tous les temps et les pourcentages mentionnés ont été mesurés ou bien simulés.

5.2.2 L'OPERATEUR DE CONVOLUTION

Le contrôle géométrique est fait en mesurant, pour chaque gélule, 64 diamètres à partir de l'image dérivée. Celle-ci est obtenue grâce à une convolution avec un masque horizontal [1 0 -1] et un masque vertical [1 0 -1], la valeur conservée étant la plus grande des deux. Cet algorithme se situe dans le deuxième processeur de l'anneau. Le temps nécessaire d'exécution au calcul de la moyenne et de l'opérateur convolution est de 2,54 ms. Ces deux algorithmes sont indissociables car le premier transmet des informations au second. Le taux d'occupation du bus interne du second processeur est de 59% tandis que celui du premier est de 69%. Dans ce cas, où le programme est partagé entre deux processeurs, la moyenne est la tâche la plus longue et elle détermine le temps total d'exécution.

Il est possible d'améliorer ces performances en faisant une meilleure répartition des tâches entre les deux premiers processeurs. Pour cela il faut partager les données de la ligne vidéo entre les deux processeurs qui effectuent chacun la moyenne et l'opération de convolution sur celles-ci, le temps de traitement est évalué alors à 2 ms et le taux d'occupation du bus interne des deux processeurs atteint 73%.

5.2.3 LE CONTROLE DIMENSIONNEL

Le troisième processeur extrait les diamètres et les transmet au quatrième qui déduit l'aire et la longueur de la gélule de chaque voie. Si une valeur anormale apparait, une demande d'interruption est faite au CPU hôte et le processeur signale par un jeton la gélule défectueuse. Le nombre d'informations à traiter ici est faible et le temps de traitement insignifiant.

5.2.4 LE CONTROLE DU REMPLISSAGE ET DE LA COULEUR

Ces contrôles sont réalisés par l'étude de deux histogrammes de niveaux de gris situés dans deux fenêtres. L'une se trouve dans la coiffe pour déterminer la couleur et l'autre dans le corps de la gélule qui est transparent afin de vérifier la présence du produit pharmaceutique. L'algorithme portant sur les six voies a été partagé entre le cinquième et le sixième processeur à cause de sa longueur. En effet, la table de liens du uPD7281est limitée à 128 et celle des noeuds à 64.

Cet algorithme porte le temps total du traitement pour une ligne à 2,6ms (fig. 5).

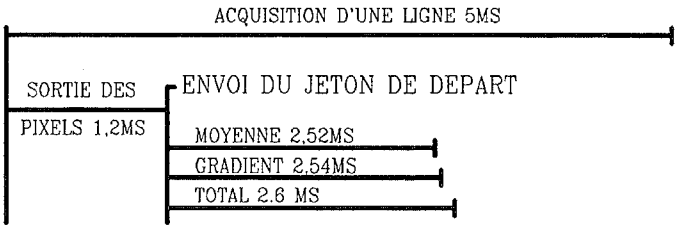

FIGURE 5

6. CONCLUSION

Nous avons vu que les performances de cette architecture sont étroitement liées à la répartition des tâches entre les processeurs. Il existe deux approches pour écrire un programme :
- partitionner la mémoire image et charger les processeurs avec le même programme
- partitionner le programme entre plusieurs processeurs quand il est trop long pour être charger dans un seul uPD7281 ou lorsqu'il est difficile de diviser la zone mémoire.

Quelque soit la configuration choisie, le temps d'exécution décroit de façon inversemment proportionnelle aux nombres de processeurs employés. La limite est imposée par un trop grand nombre de demandes simultanées d'accès à la mémoire.

Une modification intéressante de l'architecture consiste à introduire la caméra dans l'anneau où elle est vue par les autres processeurs comme un module supplémentaire. Cette configuration éviterait aux processeurs de générer les adresses sources et le système serait cadencé par les pixels de la caméra.

BIBLIOGRAPHIE

[1] D.R Slutz : The flow graph schemata model of parallel computation. MIT project MAC, TR-53, Sept 1968.
[2] J.E Rodriguez : A graph model for parallel computation. MIT project MAC, TR-64, 1969.
[3] J.Rumbaugh : A parallel asynchronous computer for data flow programs. MIT project MAC, TR-150 May 1975.
[4] D.Comte, G.Durieu, O.Gelly, A.Plas, J.C Syre : Techniques et exploitation de l'assignation unique. Toulouse : Centre d'études et de recherches de Toulouse, Vol. 9 : Final reports contract SESORI 74167, Sept. 1976.
[5]Journées d'études "Langages et machines cadencées par les données", ONERA-CERT (Toulouse), Fev. 1979.
[6] Y.M Chong : Data flow chip optimizes image processing. Computer design , Oct. 15 1984.
[7] T. Jeffery : The uPD7281 processor. Byte, Nov. 1985, 237-246.
[8] M. Chase : A pipelined data flow architecture for digital processing, the NEC uPD7281. IEEE workshop on VLSI signal processing, NEC Electronics inc., Nov. 1984.
[9] A.Zoicas : L'architecture du uPD7281, processeur à flots de données pour le traitement des images. Minis & micros n°241, Oct. 1985, 35-40.
[10] Texas instrument : TMS4461 Data sheets.
[11] Texas instrument : TMS34061 User's Guide.
[12] NEC : User's manual uPD7281.
[13] NEC : Product description uPD9305 for image pipelined processor 7281.
[14] NEC : uPD7281 Application library for image pipelined processor.
[15] R. Lecordier, P. Martin, M.Deshayes, I. Guigueno : Image processing for automated visual inspection. EUSIPCO, Sept. 1988, 319-322.