

SEPTIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 28 MAI au 2 JUIN 1979

ARCHITECTURE D'UN SYSTEME ADAPTE AU TRAITEMENT DU SIGNAL

ARCHITECTURE OF A SYSTEM DESIGNED FOR SIGNAL PROCESSING

GLINER J.M.-DOUCHIN O.-MARMEY C.-MOSSUZ C.-ROUGEOT C.

Société CROUZET SA, rue Jules Védrières, BP1014 26010 valence

RESUME

Devant faire face à un problème temps réel nécessitant :

- une puissance de calcul comparable à celle des plus gros ordinateurs actuels,
- une grande mobilité de l'installation,

nous avons été amenés à étudier une architecture matérielle capable de subvenir à nos besoins. Cette architecture fait appel aux techniques modernes que sont l'informatique distribuée et l'array processing.

Les objectifs, la conception et la réalisation d'un système souple et performant nous ont fait rejeter les logiciels généraux et par conséquent mal adaptés à ce genre de problème. Nous avons donc développé entièrement les logiciels de base nécessaires au bon fonctionnement de l'ensemble.

Il est intéressant de suivre la démarche employée, les solutions originales adoptées, la méthodologie mise en oeuvre pour résoudre nos problèmes. Il importe de cerner le progrès, les perspectives qu'une telle configuration apporte dans les domaines exigeants en temps réel que sont le traitement du signal, le traitement d'images et la reconnaissance de formes.

Conférencier : J.M. GLINER

SUMMARY

Confronted with a real time problem requiring :

- a throughput equal to that of the most powerful computers available,
- transportable hardware,

the architecture we had to design for our needs uses modern techniques such as array processing and distributed processing.

Our objective that of achieving a modular, high performance system led us to reject the idea of a general purpose system which would be inadequate to solve this kind of problem. We therefore had to rebuild entirely the basic software we needed from scratch.

Our approach to the problem, the original solutions we came up with, the methodology we employed are worth looking at in detail. Of special interest are the advantages and prospects a configuration such as this holds in the "demanding" real time field, of signal processing, image processing and pattern recognition.

Speaker : J.M. GLINER

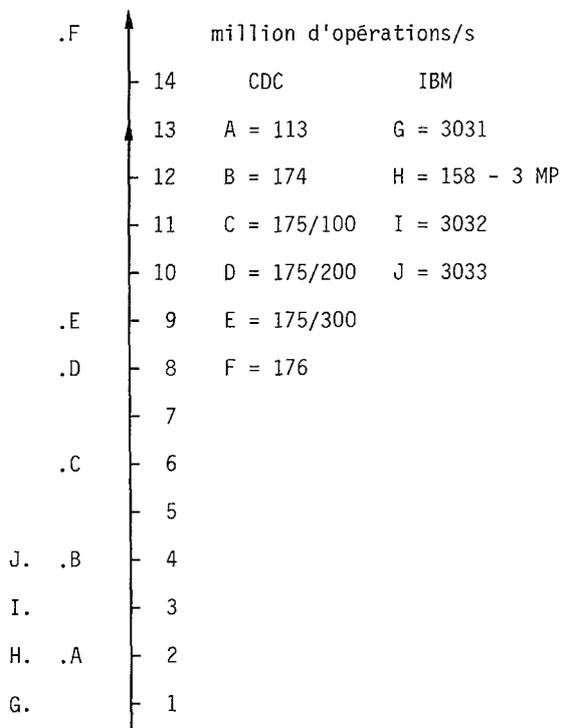


ARCHITECTURE D'UN SYSTEME ADAPTE AU TRAITEMENT DU SIGNAL
ARCHITECTURE OF A SYSTEM DESIGNED FOR SIGNAL PROCESSING

1 - INTRODUCTION

Le traitement du signal et le traitement d'image sont des techniques qui requièrent une grande puissance de calcul. Cette puissance se compte généralement en million d'opérations flottantes par seconde. Notre problème spécifique exigeait d'après des évaluations analytiques un débit de 5 millions d'opérations par seconde. Nous avons pour des contraintes de prix et d'encombrement rejeté l'emploi de très gros calculateurs. Nous nous proposons ici d'évoquer les techniques et la méthodologie que nous avons employées. Ces techniques et cette méthodologie nous ont permis de développer une solution originale.

C'est ce genre de solution qui nous semble-t-il du moins va ouvrir de nouvelles perspectives. Figure 1, Ordre de grandeur de la puissance des gros ordinateurs.



Nous voyons donc que la classe des millions d'opérations flottantes par seconde est réservée aux très gros ordinateurs, ordinateurs dont le prix restreint généralement le domaine d'activité.

2 - CHOIX D'UNE ARCHITECTURE MATERIELLE

Nous désirons donc avoir une architecture possédant les caractéristiques suivantes :

- facilité de contrôle,
- facilité d'enregistrement,
- facilité d'acquisition.

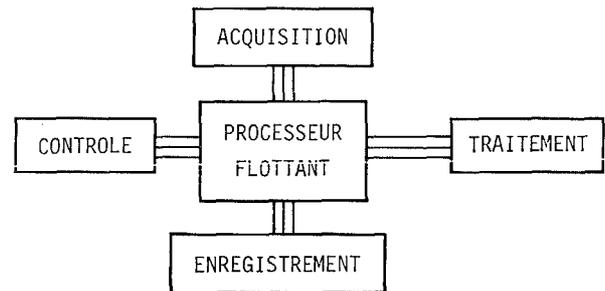
D'autre part cette architecture doit posséder les performances suivantes :

$$t \text{ multiplication} + t \text{ addition en flottant} < 0,5 \text{ ns et } *t \text{ mix} < 1 \text{ ns}$$

Il s'avère donc nécessaire d'avoir recours à un processeur spécialisé dans le calcul flottant, ce processeur étant soit indépendant soit associé à un autre ordinateur.

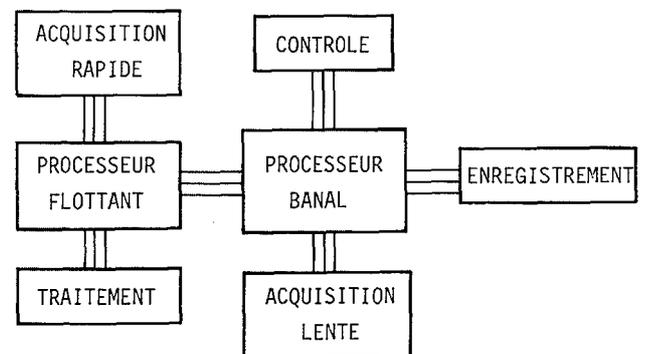
On a donc les deux solutions suivantes :

(A)



$$* t \text{ mix} = \frac{1}{3} (t + t \text{ Branch} + t \text{ Load})$$

(B)



L'avantage évident de la solution 2 réside dans :

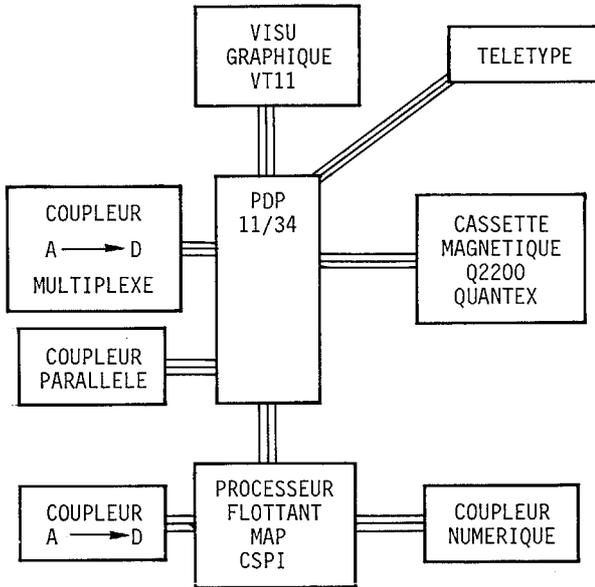
- la décentralisation des fonctions,
- l'adaptation des processeurs aux tâches auxquelles ils sont dédiés.

Un processeur flottant possède en effet des possibilités de calcul logiques, cependant la difficulté de programmation d'une telle machine est nettement supérieure à celle d'une machine conventionnelle.



ARCHITECTURE D'UN SYSTEME ADAPTE AU TRAITEMENT DU SIGNAL
ARCHITECTURE OF A SYSTEM DESIGNED FOR SIGNAL PROCESSING

Notre architecture matérielle résultante est la suivante :



PDP11/34

Cycle 1 ns

32 K mots de 16 bits

Mul. et Div. entière câblées

MAP (voir planche I)

3 bus mémoires

2 x 16 K de 32 bits à 500 nanosecondes

1 x 8 K

1 multi. + 1 addition flottante de 32 bits
= 420 nanosecondes

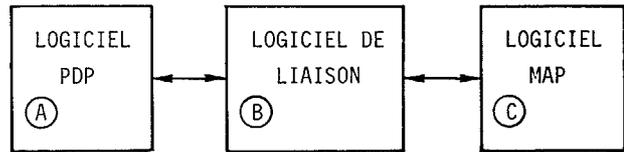
3 - CHOIX D'UNE ARCHITECTURE LOGICIELLE

Les logiciels généraux disponibles chez les constructeurs sont généralement très mal adaptés au temps réel. Leur fonction "générale" les grévant d'un temps de réponse et d'une taille mémoire mal adaptés.

C'est la raison pour laquelle nous avons été amené à développer nos propres logiciels. (Voir planche 2).

Nous avons développé 3 logiciels :

- 1 logiciel de basse PDP,
- 1 logiciel de liaison,
- 1 logiciel de base array processeur MAP.



Le coeur du système est le système A qui se compose de 4 primitives :

- activer une tâche,
- tuer une tâche,
- attendre une ressource (primitives P et V de Dijkstra)
- libérer une ressource

(Voir planches 3 et 4 pour la décomposition de ce système).

Le système B permet l'envoi de données entre A et C.

Le système C ne comporte en fait "aucun noyau".

Vu la simplicité de ce système, il est judicieux de se servir de la souplesse des interruptions logicielles

D'un point de vue réalisation nous avons tout à fait atteint nos buts.

Ayant rejeté tout logiciel de base général, il nous a été nécessaire de développer en premier lieu une méthodologie de développement de programmes :

- Méthodologie de production de programmes en moyens croisés.

Rappel : soit une machine M : soit O_m le langage exécutable et L_m un langage source.

soit une machine M' , le processeur permettant de produire O_m sur M' à partir de L_m est appelé compilateur croisé de M sur M' .

Nous disposons d'un centre de calcul donc d'une structure adaptée pour la production de programme.

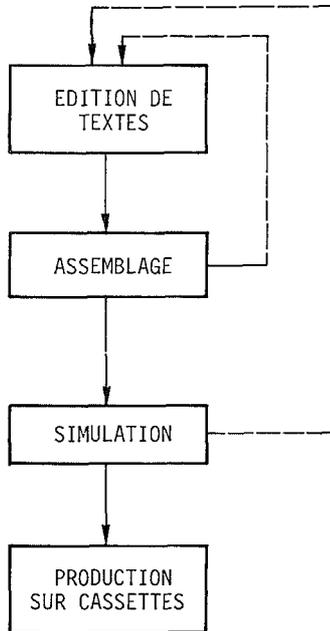
Le centre de calcul étant équipé d'un ordinateur PDP11. La compatibilité des ordinateurs de la dynastie des PDP11 nous permet de produire des programmes pour notre PDP11/34.

Pour notre array processeur, nous disposons d'un simulateur et d'un assembleur croisés fonctionnant sur notre centre de calcul.



ARCHITECTURE D'UN SYSTEME ADAPTE AU TRAITEMENT DU SIGNAL
ARCHITECTURE OF A SYSTEM DESIGNED FOR SIGNAL PROCESSING

La production de programmes se fait donc suivant le schéma suivant :



- utilisation de périphériques puissants (disque, imprimante),
- utilisation de logiciels puissants (éditeur de textes, système de gestion de fichiers).

Les cassettes produites en centre de calcul sont ensuite chargées sur notre configuration.

4 - POSSIBILITES ET PERSPECTIVE

Cette architecture offre de grandes possibilités par sa souplesse d'emploi, sa facilité de programmation, ses moyens de contrôle (console graphique), ses moyens d'enregistrement (cassette magnétique à hautes performances), sa puissance de calcul.

Nous allons énumérer de manière non exhaustive quelques unes des possibilités propres de notre configuration. Il est à noter que celles-ci peuvent être encore accrues (par l'adjonction de nouveaux périphériques, de mémoires plus rapides, par la transformation du MAP 200 en MAP 300).

Possibilités de traitement

- 1 multiplication (flottant 32 bits en 420 ns)
- + 2 additions
- FFT 1024 points réels = 7 millisecondes (2,8 avec un MAP 300).

Possibilités d'acquisition (rapide)

- coupleur numérique 16 bits à 1 megahertz
 - coupleurs analogiques 2 voies à 125 kilohertz
- Chacune avec conversion analogique digital à la volée.

Possibilité d'enregistrement

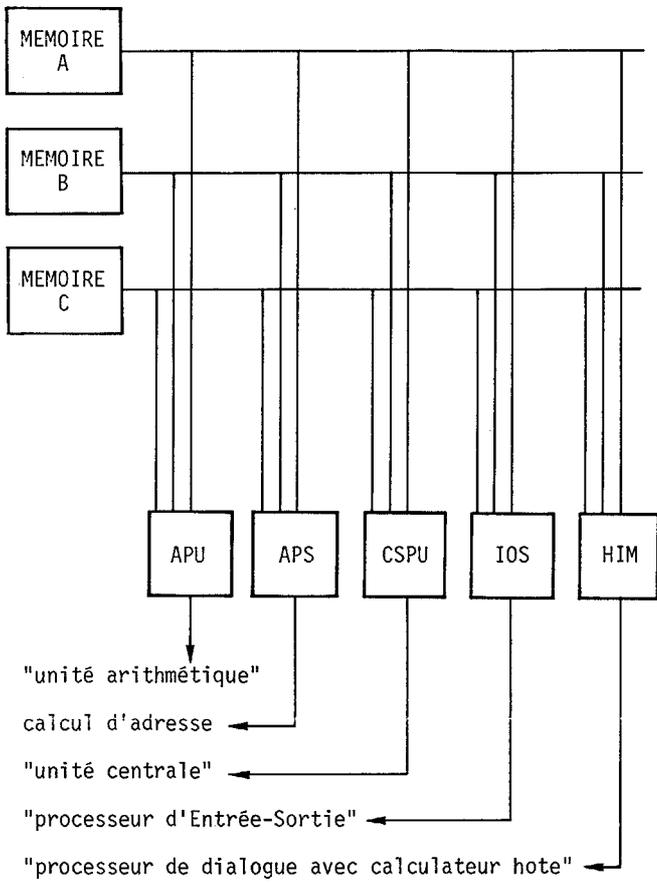
Temps apparent d'accès à la mémoire 160 nanosecondes
cassettes $3 \cdot 10^3$ octets par seconde avec une capacité de $3 \cdot 10^6$ octets par unités.

Possibilité de contrôle

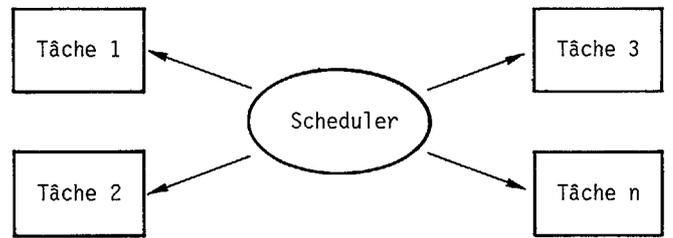
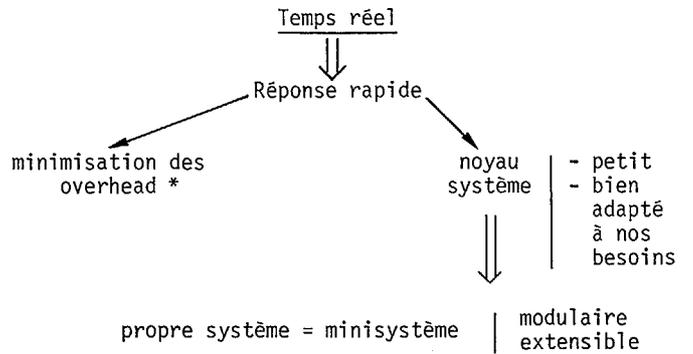
Console de visualisation permettant de visualiser 1500 vecteurs (ex. utilisation en oscilloscope).

On constate donc que ces performances offrent de nouveaux horizons et qu'il serait dommage de ne pas avoir l'oeil ouvert sur les possibilités offertes en traitement du signal et en traitement d'image.

ARCHITECTURE D'UN SYSTEME ADAPTE AU TRAITEMENT DU SIGNAL
 ARCHITECTURE OF A SYSTEM DESIGNED FOR SIGNAL PROCESSING



- Planche 1 -



Scheduler = coeur du système

- Lancer les tâches avec des priorités différentes. Les synchroniser par évènements.
- Attribuer les ressources du système aux différentes tâches activées

Scheduler

- Gestion de l'Unité Centrale
- Gestion des ressources

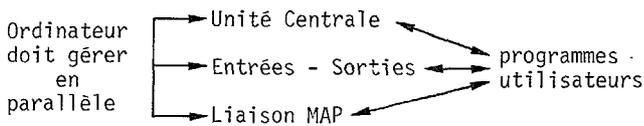
- Planche 2 -

* overhead : temps pris par le système pour gérer les ressources.

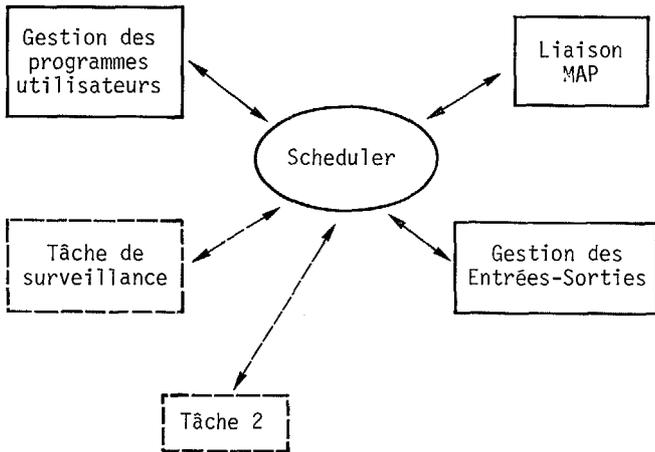


ARCHITECTURE D'UN SYSTEME ADAPTE AU TRAITEMENT DU SIGNAL
 ARCHITECTURE OF A SYSTEM DESIGNED FOR SIGNAL PROCESSING

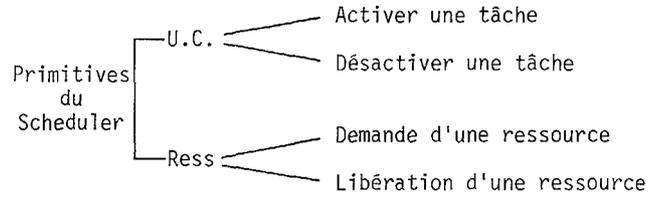
Entrées - Sorties // Utilisation de l'U.C. par une tâche (sauf début et fin)



DECOMPOSITION DU SYSTEME

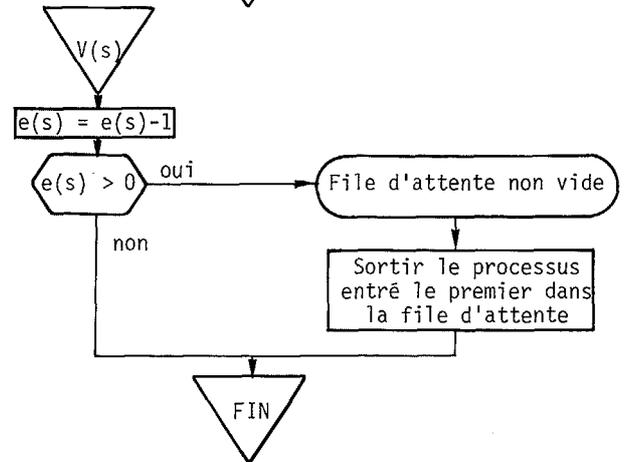
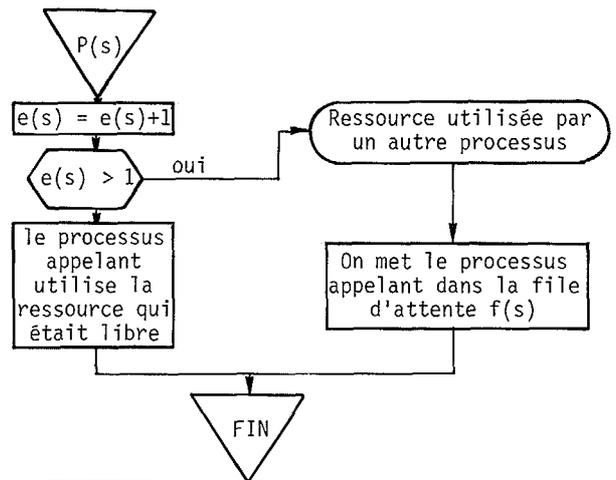


- Planche 3 -



GESTION DES RESSOURCES

Primitives de Dijkstra P, V. } ≤
 Semaphore $e(s) = e_0(s) \leq 0$
 File d'attente $f(s)$ $f_0(s)$ vide



- Planche 4 -