

NEUVIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 16 au 20 MAI 1983

PROCESSEUR TEMPS REEL DE TRAITEMENT DE SIGNAL (PTTS)
REAL TIME PROCESSOR FOR SIGNAL PROCESSING ALGORITHMS EMULATION

C. HANRAN - M. LEVY

CIT-ALCATEL- Département Transmission- Direction Télématique- Centre de Villarceaux-91620-LA VILLE DU BOIS-
(FRANCE) -

RESUME

Résumé :

Le Traitement numérique de signal, notamment adaptatif est de plus en plus souvent introduit dans les équipements de communication, de transmission (Egaliseurs adaptatifs, annuleurs d'échos, etc...).

Une démarche courante lors de l'étude d'un produit industriel consiste à modéliser mathématiquement le traitement de signal à effectuer, puis à simuler ce traitement sur ordinateur, et enfin à confirmer les résultats de simulation par une maquette de faisabilité.

La démarche des auteurs a été de vouloir réduire les délais et les coûts d'une telle étude, particulièrement les simulations souvent longues et tributaires du modèle mathématique choisi.

Dans ce but, il a été conçu et réalisé un processeur de signal programmable dont les caractéristiques sont les suivantes :

- Emulation en temps réel et mise en oeuvre sur site d'algorithmes mathématiques.
- Possibilité de modification des algorithmes par programmation.
- Possibilité de reconfigurer les dimensions des mots binaires.
- Possibilité d'exploitation statistique des résultats sur un ordinateur hôte.

Il a été conçu deux versions de ce processeur, dont les modules sont interchangeables; l'une pour mise au point en laboratoire, l'autre pour exploitation sur site.

Les deux versions sont opérationnelles et utilisées pour diverses applications.

SUMMARY

Summary :

Digital signal processing, especially adaptive, is introduced more and more often in modern transmission and communication equipments (adaptive equalizers, echo cancellers, etc...).

The design of an industrial equipment is usually composed of the following steps : first a mathematical modelization of the signal processing desired, then computer simulations, and finally a confirmation of the simulation results by a feasibility mockup.

The authors approach has been to reduce the cost and the time conception of such a design, and especially computer simulations which often sharply depend of the mathematical model.

Hence a programmable signal processor has been designed and implemented; it has the following characteristics :

- Real time emulation and in field test of mathematical algorithms.
- Software modification of the algorithms and selectable word-length.
- Statistical analysis of results by a host computer.

Two versions of the processor were designed, with high modularity :

One for laboratory investigations, the other for use in the field. Both are operational and used for different applications.



PROCESSEUR TEMPS REEL DE TRAITEMENT DE SIGNAL (PTTS)
REAL TIME PROCESSOR FOR SIGNAL PROCESSING ALGORITHMS EMULATION

I - GENERALITES et PHILOSOPHIE

Le traitement numérique du signal a connu un développement important ces dernières années et est utilisé maintenant dans de nombreux équipements industriels, civils ou militaires, en particulier dans le domaine de la communication (transmission numérique, transmission de données, traitement de la parole). Ce développement est dû, en grande partie, aux progrès de la technologie qui permet d'effectuer des traitements sophistiqués notamment, en filtrage fixe ou adaptatif en temps réel.

Ces progrès technologiques ont favorisé l'introduction et le développement de deux types de microprocesseurs.

Le premier comme processeur hôte, gérant des tâches d'exploitation; le second microprogrammable effectuant les parties numériques du traitement des signaux.

Lorsque l'on étudie un nouvel équipement nécessitant une part importante de traitement de signal, les différentes phases de l'étude peuvent être schématisées de la façon suivante :

a) Etude Théorique : elle définit les algorithmes à mettre en oeuvre, après avoir modélisé les signaux à traiter.

b) Etude Informatique : elle simule les signaux à traiter ainsi que le ou les traitements définis dans l'étude théorique.

c) Maquette de Faisabilité : les principes de réalisation étant acquis, on met en oeuvre une première version expérimentale.

d) Prototype d'industrialisation : Cette phase représente l'aboutissement de la faisabilité et la préparation à la production en série.

Il est légitime de vouloir réduire les coûts et durées d'études et, pour cela, il faut remarquer que les phases a) et d) sont difficilement réductibles. La phase b) est nécessaire mais souvent longue et coûteuse, en particulier lorsque l'on veut optimiser finement le traitement. De plus, elle présente souvent le défaut d'être soumise à la validité du modèle de simulation utilisé.

La phase c) peut être d'une durée et d'un coût raisonnable; elle débouche à sa fin sur des essais réels et donc sur une sanction de trois phases a), b), c).

Durant la phase C) il peut arriver que des retours à la phase b) soient nécessaires, lorsque l'on a sous dimensionné les organes de calcul par manque de moyens ou de temps, ce qui peut pénaliser l'étude.

Il est possible d'envisager aujourd'hui une articulation différente du déroulement des études :

a') Etude Théorique : éventuellement complétée de simulations sommaires.

b') Simulations sur un processeur numérique en temps réel, c'est-à-dire une maquette de faisabilité programmable permettant de dimensionner correctement les organes de traitement numérique et de vérifier en temps réel les principes de traitement définis en a').

d') Prototype d'industrialisation : idem que d).

La phase b) est une contraction des phases b) et c) du premier scénario. Elle nécessite de disposer d'un processeur numérique ayant des possibilités générales, de manière à l'adapter à de nombreuses applications. Les points essentiels sont :

- . Rapidité des organes de calculs

- . Dimensionnement des organes de mémoires et de calculs

- . Souplesse de programmation

- . Souplesse des organes d'entrées-sorties

Avec un tel système, on peut :

- dimensionner avec précision les organes de traitement du prototype industriel (format des mots binaires, par exemple).
- analyser le comportement en temps réel des algorithmes envisagés.

La validité des modèles mathématiques est rapidement vérifiée, et, si un complément d'étude théorique est nécessaire en parallèle, il peut aisément être fait.

Les "grosses simulations" ne sont plus nécessaires. L'ensemble des dépenses b) et c) est remplacé par un investissement en matériel consacré au développement, et à la conception du processeur de traitement de signal.

Pour certaines applications, il n'est pas envisageable de déplacer l'ensemble des outils sur le site de l'expérimentation durant la phase b'). Pour pallier cet inconvénient, il suffit de disposer d'une version "mobile" de la maquette programmable. L'essentiel de la conception est fait en laboratoire. Lorsque le programme est au point, celui-ci est stocké sur une mémoire et implanté sur la version "mobile".

Le paragraphe II décrit l'architecture d'un processeur numérique qui a été étudié et mis en oeuvre à la CIT-ALCATEL.

Les applications visées sont de manière non limitative :

- . La reconnaissance et la synthèse de parole
- . Les modems à haute rapidité
- . La transmission de parole à débit réduit.
- . La définition de processeurs de signaux intégrés (VLSI).

Le paragraphe III décrit le langage utilisé pour programmer cette machine et un exemple d'application : un filtre transversal adaptatif.

Il illustre les possibilités du processeur.

Le paragraphe IV tire quelques conclusions de l'étude, et donne quelques perspectives.

II - ARCHITECTURE

Le but visé est triple :

- Disposer d'un processeur microprogrammé orienté traitement de signal avec le répertoire d'instructions adapté aux besoins de chaque application.
- Avoir un outil destiné aux utilisateurs non initiés à la microprogrammation, permettant de faire la mise au point du logiciel.
- Offrir la possibilité d'émuler en temps réel un processeur de signal intégré pour évaluer ses performances.

1 - Structure générale du Processeur du Signal

Ce processeur est orienté traitement de signal notamment en filtrage fixe ou adaptatif. Il vise le traitement des signaux dont la fréquence d'échantillonnage se situe autour d'une dizaine de kilohertz.

Pour obtenir les performances requises, le PTTS a été conçu autour du microprocesseur en tranche de la famille 2900. Son architecture est basée essentiellement sur celle d'une machine du type pipe-line.

Le PTTS est organisé autour d'un processeur microprogrammé. Il est composé de :

- Un compteur ordinal constitué d'une unité de calcul d'adresses et d'une pile associée.
- Une mémoire programme constituée d'une mémoire morte de programme et des données nécessaires (des constantes) au Processeur Arithmétique.
- Un processeur microprogrammé.

La figure (1) représente une configuration typique dans laquelle le processeur est inséré entre une chaîne d'acquisition et un calculateur.

2 - Description du processeur microprogrammé fig. (2)

Son architecture est basée sur le critère du parallélisme des fonctions. Elle est donc du type pipe-line. Il comprend :

- Un ensemble de mémoires de données
- Un processeur arithmétique en virgule fixe.
- Un processeur arithmétique en virgule flottante.
- Deux bus de données indépendantes.
- Un organe d'entrée-sortie
- Une unité de commande microprogrammée.

2.1 - Mémoire de données

Deux blocs mémoires à accès aléatoire (RAM) sont utilisés. La mémoire de données a un temps d'accès inférieur à 90 ns, et une capacité de 1.024×24 bits. Elle est extensible à 2.048×24 bits.

Les 2 blocs sont symétriques et jouent le rôle de mémoires bloc-notes. Ils servent à stocker des échantillons, des coefficients et des données temporaires utilisées par les Processeurs Arithmétiques.

A chaque bloc mémoire (MEMX, MEMY) est associée une unité de calcul d'adresse qui offre des facilités d'adressage notamment lors de l'exploration d'un tableau.

Chaque unité de calcul d'adresse comprend :

- 1 registre d'adresse : AX
(ou AY pour la mémoire Y)
- 4 registres d'index : IX, IX0, IX1, IX2
(ou IY, IY0, IY1, IY2)

Le premier registre IX (IY) n'est pas accessible à l'utilisateur. Il est réservé à l'usage interne et chargé à ± 1 . Au cours de l'exploitation d'un tableau, l'adresse AX de la mémoire X peut s'incrémenter de ± 1 , $\pm IX0$, $\pm IX1$, $\pm IX2$.

$$AX = AX \pm IXi \quad \text{avec} \\ 2 \leq IXi < 2047$$

Puisque l'unité de calcul d'adresse de la mémoire Y est indépendante de celle de X, on a également :

$$AY = AY \pm IYi \\ 2 \leq IYi < 2047$$

2.2 - Processeurs Arithmétiques

2.2.1 - Processeur arithmétique virgule fixe (PFI)

Le PFI comprend 3 opérateurs :

- le multiplicateur-accumulateur,
- l'unité arithmétique et logique,
- l'opérateur de décalage programmable.

Les opérandes peuvent être tronquées de manière à adapter la longueur des mots machine aux applications spécifiques.

2.2.1.1 - Le multiplicateur-accumulateur

C'est un multiplicateur-accumulateur en LSI 16x16 bits, résultat sur 35 bits dont 4 pour la redondance du signe.

- Le multiplicateur

Le multiplicande et le multiplicateur peuvent être tronqués indépendamment bit par bit. Leur dimension peut être programmée de 8 à 16 bits.

- L'accumulateur A :

L'accumulateur A garde la précision de 35 bits pendant l'opération du type somme de produits.

Le résultat final est ramené à 24 bits lors de son transfert sur les bus X et Y.

Il existe également la possibilité de tronquer les 24 bits du résultat jusqu'à 16 par programmation.

2.2.1.2 - L'unité arithmétique et logique/accumulateur B

Cette unité arithmétique et logique, indépendante du multiplicateur-accumulateur, augmente le degré de simultanéité des opérations arithmétiques et logiques du PTTS. Elle contient également un accumulateur de 24 bits.

La même possibilité de troncature est prévue. Par programmation, les opérandes peuvent être tronquées de 0 à 16 bits par pas de 2. Aussi, le format de l'accumulateur passe-t-il de 24 à 8 bits.



Grâce à la symétrie de cet opérateur, on peut indifféremment effectuer les calculs sur les données présentes sur le bus X ou sur le bus Y.

2.2.1.3 - Opérateur de décalage à droite

Cette unité réalise l'opération de division par 2^n .

Le décalage à droite est programmable de 0 à 16 bits et s'effectue en un seul cycle-machine.

La constitution de cet opérateur programmable diffère de celle utilisée couramment dans les processeurs classiques pour lesquels le temps d'exécution est égal à $n \times \text{cycle-machine}$.

2.2.2 - Processeur arithmétique en virgule flottante (PFL)

Ce processeur a été conçu à partir d'un VLSI pour effectuer des analyses statistiques (moyenne, variance,...) et pour engendrer des fonctions trigonométriques et exponentielles. Le PFL effectue des tâches spécifiques d'analyse du traitement du PFI.

2.3 - Bus de données

Pour atteindre la rapidité nécessaire au traitement, les bus de données sont au nombre de 2 : le bus X et le bus Y.

La séparation du chemin de données en 2 voies distinctes offre un accès direct et simultané des opérands aux unités de calcul et à la mémoire bloc-notes.

2.4 - Organe d'entrée-sortie

L'organe d'entrée-sortie est composé de 2 unités :

- le canal d'entrée-sortie à accès direct couplé avec la chaîne d'acquisition analogique/numérique (débit maximum : 220 ns/mot de 16 bits).
- le canal d'entrée-sortie programmé, qui permet le couplage avec un calculateur hôte.

2.5 - Unité de commande microprogrammée

L'unité de commande est la partie la plus importante du processeur de signal. C'est celle qui commande le processeur arithmétique et supervise les activités des bus évitant ainsi le conflit d'accès. C'est celle qui règle la synchronisation des événements extérieurs vis-à-vis de la machine.

La microprogrammation comprend essentiellement sur le plan matériel :

- Un séquenceur
- Une mémoire microprogramme
- Un registre de micromots

2.5.1 - Séquenceur

Le séquenceur fournit 12 bits d'adresse. Sa capacité d'adressage est donc de 4 Kbits.

2.5.2 - Mémoire de microprogramme

Dans une configuration figée, cette mémoire est une PROM. Lorsque le répertoire d'instructions est susceptible d'évaluer selon les applications, il est préférable d'utiliser une mémoire vive.

La configuration actuelle fait appel à une mémoire RAM dont la capacité est de 2 k x 128 bits.

La largeur de 128 bits du micromot est nécessaire étant donné :

- le choix de type de codage des micro-instructions
- le parallélisme des Processeurs Arithmétiques.

2.5.3 - Registre micromot

Les micro-instructions contenues dans le registre micromot remplissent les fonctions suivantes :

- fournir au séquenceur l'adresse de la prochaine micro-instruction.
- fournir des commandes nécessaires aux différents blocs de la machine.

2.6 - Structure de la micro-instruction

La structure parallèle des opérateurs arithmétiques et la recherche d'une grande simultanéité de leur exécution conduisent à un découpage du micromot en plusieurs champs.

Le type de codage utilisé est le codage par champs indépendants, qui allie la simplicité à la puissance d'exécution, puisque plusieurs tâches distinctes peuvent être effectuées en même temps.

La micro-instruction est formée de 6 champs distincts liés aux éléments suivants :

- séquenceur
- organe de calcul d'adresse
- opérateurs arithmétiques et logiques
- bus
- opérateur d'entrée-sortie.
- données littérales.

3 - Performances

Les performances sont identiques quel que soit le nombre de bits tronqués (ex : le passage 16 bits vers 8 bits des opérands du multiplicateur).

- temps d'exécution d'un micro-instruction : 220 ns.

- opération du type "lecture, multiplication-accumulation" : 220 ns

- opération du type "lecture, multiplication-accumulation, écriture, saut" : 440 ns.

Les performances permettent d'envisager de traiter avec le , de nombreuses applications.

III - EXEMPLE D'APPLICATION : Filtre Transversal Adaptatif

On considère un filtre à 256 coefficients asservi selon un algorithme du gradient stochastique (Annuleur d'Echo).

Si (x_k) est la suite numérique des signaux d'entrée, le filtre de coefficients $(f_i)^{(k)}$ calcule :

$$\hat{y}_k = \sum_{i=0}^{255} f_i(k) x_{k-i}$$

Puis l'erreur e_k par rapport à une référence y_k :

$$e_k = y_k - \hat{y}_k$$

Les coefficients sont ensuite asservis selon l'algorithme :

$$f_i(k+1) = f_i(k) + \mu e_k x_{k-i}$$

μ est une constante positive.

Le programme qui réalise ces opérations s'écrit en deux instructions :

```

FILTER :   IF AX = 0 GOSUB AXMOD
           & BX = MEMX, BY = MEMY
           & U = BX, V = BY, W = W + U*V
           & F = BY, G = F + S XOR E ;
    
```

```

IF AY = 0 GOSUB @CTR, FILTER
& BY = G
& MEMY = BY
& AX = AY - IX0, AY = AY - IY0 ;
    
```

IV - CONCLUSIONS ET PERSPECTIVES

Le PTTS, dont les caractéristiques principales ont été décrites, se révèle être une machine de traitement de signal bien adaptée aux applications temps réel, telles que l'annulation d'écho, l'égalisation adaptative, l'élimination de bruit, ainsi que la réduction de débit.

Il est utilisé aujourd'hui pour différentes études, et ses versions "mobiles" ont permis, pour certaines applications, de faire des essais sur site qui ont validé les investigations de laboratoire.

L'exploitation de cette première génération de machine temps réel permettra d'aborder d'autres applications, notamment dans le domaine du traitement des images numérisées.

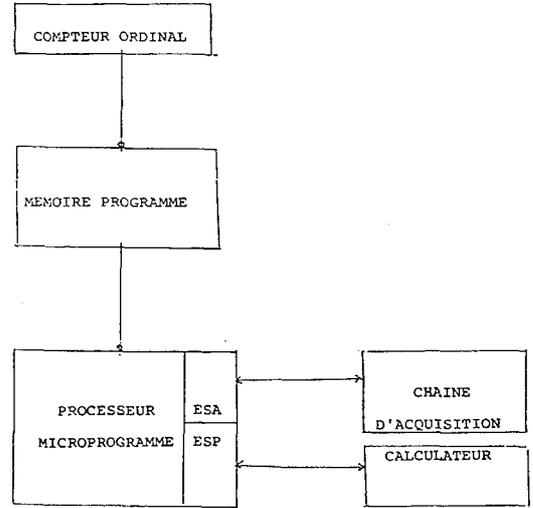


Figure 1

PROCESSEUR COUPLE AU CALCULATEUR

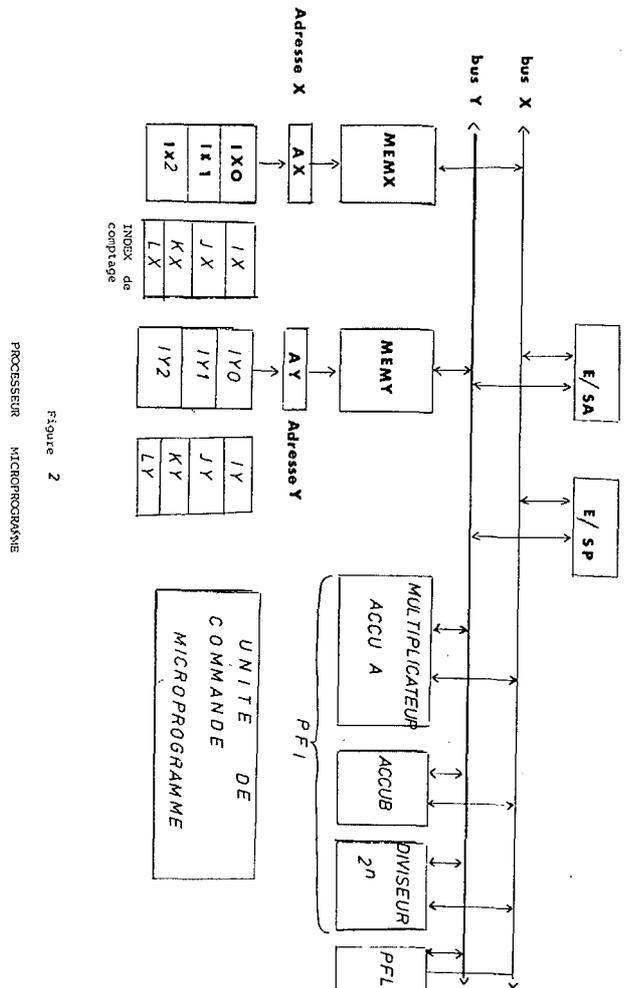


Figure 2

PROCESSEUR MICROPROGRAMME

