

NEUVIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 16 au 20 MAI 1983

MACHINE DE TRAITEMENT DE SIGNAL ADAPTEE AU CAS DE RECEPTEURS A
ANTENNE DE REVOLUTION

Jean PASTERNAK

THOMSON-C.S.F., Division A.S.M., Chemin des Travaux, BP 53 06801 CAGNES/MER CEDEX

RESUME

Nous présentons ici une architecture de machine particulièrement adaptée au traitement des signaux d'un récepteur sonar actif.

Cette machine exploite le fait que chacun des signaux hydrophoniques subit les mêmes transformations et que les algorithmes de traitement ne dépendent pas du signal lui-même. De même, la base acoustique ayant une symétrie de révolution, il est possible d'élaborer simultanément toutes les voies avec la même séquence de calcul.

La machine est constituée par un ensemble d'opérateurs identiques traitant en parallèle une ou plusieurs voies. Les opérateurs sont pilotés par un séquenceur unique programmable.

Un organe de communication sous forme de "bus en anneau" permet d'échanger les signaux entre opérateurs pour réaliser l'opération de formation des voies de manière efficace.

Après avoir justifié le choix d'une telle architecture, nous décrivons les différents organes de la machine ainsi que les logiciels de base nécessaires à l'écriture et mise au point des programmes opérationnels.

SUMMARY

We present here a structure which particularly suits to active sonar signal processing.

This machine benefits by the fact that all the hydrophone signals are processed identically and that the algorithms which apply do not depend of the signal itself. Furthermore the acoustic array being supposed to have a revolution symmetry it is also possible to form all the beams simultaneously with the same calculation sequence.

The machine is made of a set of identical operators which are able to parallel process one or several channels. The operators are driven by only one programmable sequencer.

A communication link is provided under the form of a cyclic shift register bus. The signals are then exchanged between the different operators so as to fulfil the beam forming in a efficient way.

After an explanation about the main objectives of such a structure, we describe the different sub-assemblies of the machine and we present the basic softwares which were necessary to write and to test the operational programs.



INTRODUCTION

L'accélération technologique des moyens de traitement numérique a permis progressivement de substituer aux chaînes analogiques des chaînes numériques plus performantes et plus fiables. Si bien qu'à ce jour, en traitement sonar, les circuits analogiques associés à un capteur hydrophonique se limitent aux préamplificateurs et filtres de pré-échantillonnage avant numérisation.

Néanmoins, le volume d'opérations arithmétiques associé au débit d'information que l'on trouve en amont des chaînes de traitement est tel qu'il n'a pu donner lieu qu'à des réalisations câblées spécifiques de chaque application.

Cette démarche, bien qu'optimale sur le plan de la réalisation, ne l'est pas d'un point de vue industriel puisqu'elle entraîne un foisonnement d'études et de matériels difficile à reconduire d'un équipement à l'autre.

Une réponse satisfaisante à ce problème se trouve dans les solutions programmées où le même matériel permet de résoudre plusieurs classes de problèmes. Mais à ce jour, les calculateurs standard n'ont pas les performances suffisantes pour être appliqués au traitement du signal en amont des chaînes de réception.

Le but de cet exposé est de présenter une réalisation de processeurs programmables orientée sonar actif, dont les performances en capacité de calcul pour un encombrement raisonnable permettent d'envisager des solutions programmées pouvant rivaliser avec les solutions câblées.

CHOIX DE L'ARCHITECTURE

Si l'on examine les traitements effectués sur les signaux de réception d'un sonar actif classique, on constate qu'il existe certaines symétries dont on peut tirer partie :

- Une symétrie des traitements (Fig. 1) :

Chaque capteur, chaque voie formée, subissent le même traitement. Ce traitement est indépendant du signal lui-même et fait appel à des opérations arithmétiques simples (produit et somme).

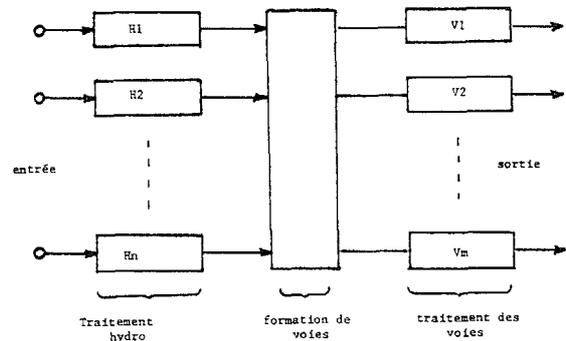


FIG. 1

- Une symétrie spatiale (Fig. 2)

L'antenne acoustique a une symétrie de révolution (cylindrique). La répartition des capteurs à la périphérie de la base est régulière. Les voies formées sont équiréparties sur l'horizon avec un pas angulaire constant. Le nombre de voies formées est un multiple du nombre de capteurs. Chaque voie est donc formée de manière identique.

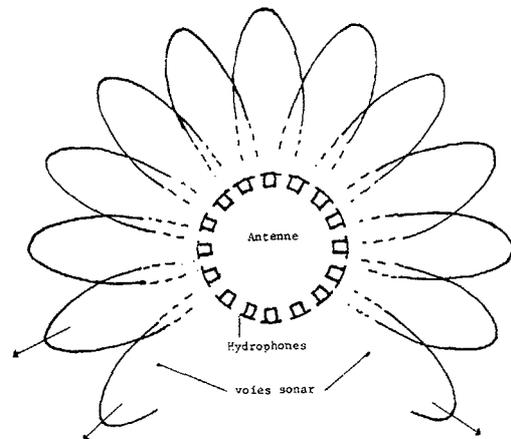


FIG. 2

Le concept de la machine de traitement est basé sur cet aspect de symétrie. Chaque capteur et chaque voie sont traités dans des opérateurs semblables conçus pour effectuer les opérations arithmétiques de base. Les calculs sont pilotés de manière synchrone par un séquenceur unique contenant le programme à exécuter. On obtient ainsi une architecture de type SIMD (Single Instruction Stream, Multiple Data Stream). Pour effectuer le calcul de formation de voies, les opérateurs doivent communiquer entre eux. Ils sont reliés deux à deux par l'intermédiaire d'un registre tampon formant ainsi un bus en anneau par lequel tous les opérateurs peuvent échanger des données de manière synchrone. Cette disposition permet de combiner les signaux d'hydrophones adjacents

et former en un seul temps de calcul l'ensemble des voies sonar. Le Bus est relié en un point de l'anneau à un organe d'entrée-sortie connecté sur le bus d'un microprocesseur hôte du type 68000. C'est par ce canal que le calculateur hôte prélève les résultats de fin de traitement et peut transmettre aux opérateurs certains arguments nécessaires aux calculs. (Fig. 3).

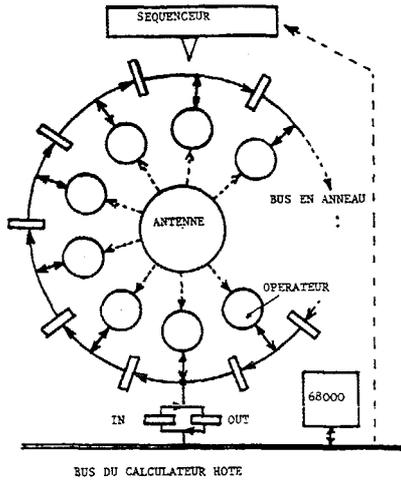


FIG. 3

L'OPERATEUR

Les opérations de traitement de signal à effectuer sont toutes du type produit, somme et somme de produits sur des données réelles ou complexes. Il s'agit de filtrage, corrélation, formation de voie, analyse spectrale par FFT. Les ressources suffisantes à la réalisation de ces algorithmes peuvent être représentées par la Fig. 4.

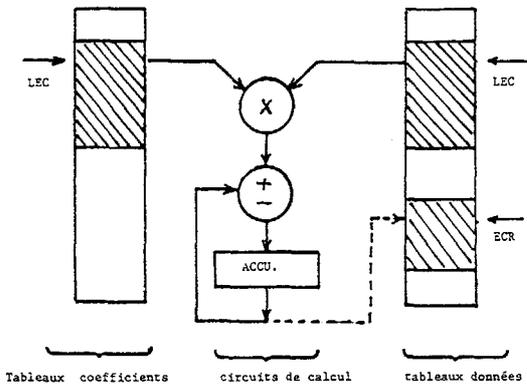


FIG. 4

Les données sont rangées dans une mémoire privée appartenant à l'opérateur. Toutes les mémoires reçoivent la même adresse issue du séquenceur, tandis qu'un bus issu d'une mémoire unique fournit les coefficients de pondération communs à tous les opérateurs.

Un multiplieur accumulateur du type TRW 1010 constitue l'unité arithmétique de l'opérateur. Un bus interne relie l'unité arithmétique, la mémoire des données de capacité 4K mots complexes, le bus en anneau connecté aux opérateurs voisins et un registre série-parallèle relié au convertisseur numérique d'entrée.

La mise en oeuvre du strict minimum de circuits nécessaires aux calculs permet d'obtenir un opérateur compact qui occupe une demi-carte au format double Europe (Fig. 5).

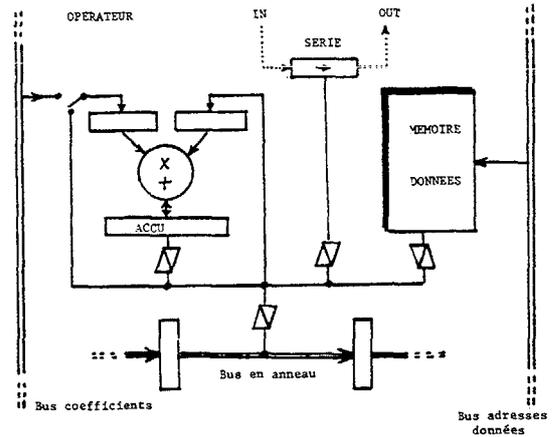


FIG. 5

LE SEQUEUR

C'est un automate programmable essentiellement constitué de mémoires mortes reprogrammables (REPROM) de capacité 4K mots de 8 bits dont le choix peut être justifié ainsi :

- Le traitement du signal dans la chaîne de réception ne comprend qu'un nombre restreint de modes et on n'envisage pas de reconfigurer les traitements durant la vie du matériel.
- Le temps de cycle de l'opérateur est suffisamment grand pour s'accommoder du temps d'accès d'une mémoire REPROM (200 ns), sous réserve qu'à chaque micro-instruction, on soit capable d'élaborer les micro-commandes de l'opérateur et le calcul des deux adresses Données et Coefficient.



- L'usage de mémoire REPRON permet de minimiser le coût et le volume du matériel de mise au point de programme puisqu'elles sont réutilisables à chaque modification. Par souci d'efficacité, le séquenceur est réduit à sa forme la plus simple. Il peut se décomposer en trois parties :

- . une mémoire de microprogramme,
- . un système de séquencement,
- . deux microséquenceurs satellites chargés du calcul d'adresse.

La mémoire de microprogramme d'une capacité de 4 Kbits délivre les microcommandes sur une largeur de 80 bits. Le format est unique et contient tous les champs affectés à chacune des parties de la machine. On peut ainsi, à chaque cycle élémentaire, définir :

- Le mode d'accès à la micro-instruction suivante,
- Les commandes des deux séquenceurs d'adresse,
- Les commandes de l'ensemble des opérateurs,
- Les dialogues entre la machine et le calculateur hôte.

Le séquencement s'effectue suivant le procédé d'adresse suivante. La micro-instruction en cours délivre in-extenso l'adresse de la micro-instruction à venir. En cas de test, l'adresse suivante est altérée par le résultat du test qui se substitue au faible poids d'adresse. L'adresse suivante est mémorisée dans un registre à chaque cycle. Ce registre est matérialisé par l'une des cases d'une mémoire de 16 mots qui, adressée par un compteur-décompteur, constitue une pile. Cette pile permet de sauvegarder l'adresse courante lors d'interruptions ou d'appels sous-programme.

Un dispositif en logique combinatoire synthétisée dans un réseau FPLA est chargé de prendre en compte les interruptions issues de l'environnement extérieur (arrivée d'échantillons) ou du calculateur hôte (dialogues opérationnels).

L'accès aux programmes d'interruption et les déroutements orchestrés par le calculateur hôte s'effectuent grâce à une mémoire annexe qui permet de modifier l'adresse de la micro-instruction courante (registre d'éclatement).

Les adresses des mémoires de donnée et de coefficient sont élaborées par deux automates identiques contrôlés par le microprogramme. La nature des traitements affectés à la machine peut amener des simplifications au problème général du calcul d'adresse :

- Le nombre de tableaux de données et coefficients est fini et relativement peu élevé,
- Les configurations de traitement sont limitées et figées pour un matériel donné,
- L'exploitation des tableaux donne lieu à des lois d'adressage simples. En général il s'agit d'incrémenter l'adresse avec un pas constant.

Ces considérations amènent à penser que les séquences d'adresse peuvent être inscrites une fois pour toutes dans une mémoire morte plutôt que de faire appel à une circuiterie à base d'opérateurs arithmétiques. C'est ce qui a été retenu dans la présente réalisation.

Une banque de registres (pointeurs d'adresse) contient les adresses courantes de tableau. Le contenu de l'un des registres est dirigé vers le bus d'adresse général et en même temps sur les entrées adresse d'une mémoire morte. A cette adresse de la mémoire est inscrite la prochaine adresse venant naturellement dans la suite du calcul en cours. La nouvelle valeur est inscrite dans le registre à la fin du cycle et devient disponible pour le pas de calcul suivant. Le processus se poursuit pour tous les éléments du tableau à traiter. La dernière adresse du tableau génère par la mémoire morte un indicateur de fin de séquence permettant d'arrêter le calcul.

Ce mécanisme ne peut convenir à toutes les applications. Sa limitation est due au fait que la loi d'adressage est unique au sein d'un tableau puisqu'à une entrée de la mémoire des adresses ne peut correspondre qu'une sortie. Il faudrait donc autant de mémoires qu'il y a de lois d'accès pour exploiter les données d'un tableau.

Pour résoudre ce problème tout en conservant le même principe de génération d'adresse, il a fallu abandonner la numération classique. En exprimant les adresses mémoire dans un système de numération par résidus, il est possible de représenter un nombre quelconque par plusieurs autres plus petits sans liens matériels entre eux.

Les adresses mémoires dans la machine étant codées sur 12 bits, si l'on prend une base résidus en 2^n , 2^{n-1} avec $n=6$, ces adresses peuvent être fractionnées en deux mots indépendants de six bits.

Les lois d'adressage sont codées dans deux mémoires mortes affectées à chacun des modules de la base. Les six bits d'adresse restants permettent de sélectionner à la fois la zone d'adresse et les lois d'incrément de chaque tableau.

Les conséquences pratiques de cette disposition sont que le champ d'adressage n'est plus que de $64 \times 63 = 4032$ mots au lieu de 4096, et que les données d'un tableau ne sont pas rangées dans un ordre naturel, mais éparpillées dans la mémoire suivant la façon dont s'exprime l'adresse dans la base résidus choisie.

Le synoptique du séquenceur est représenté Fig. 6.

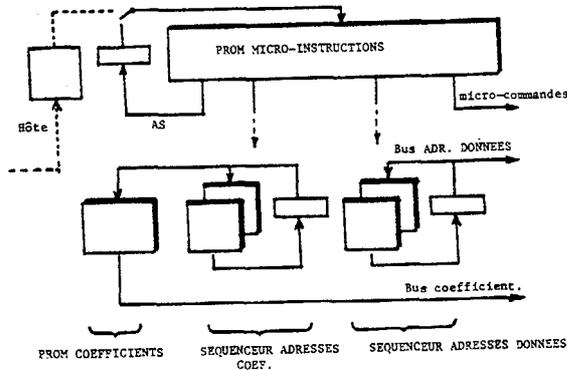


FIG. 6

LE LOGICIEL

Malgré une réalisation matérielle simple, on ne peut envisager de programmer et mettre au point la machine sans une aide logicielle appropriée.

La chaîne de production de programmes est une chaîne croisée écrite en langage PASCAL sur EXORMAX (système Motorola à base de 68000). Elle comprend :

- Un assembleur de microprogrammes,
- Un simulateur machine décrivant la réalisation Hardware,
- Un simulateur fonctionnel pour la mise au point de programme,
- Des outils de mise au point sur site implantés dans le calculateur hôte.

L'ASSEMBLEUR

Contrairement aux habitudes passées, on a voulu créer un langage clair qui s'apparente à un langage de haut niveau tout en restant très proche de la réalité physique de la machine.

Le programmeur peut ainsi contrôler finement l'exécution de ses algorithmes en utilisant les phrases d'un langage naturel. L'écriture, la mise au point et la documentation des programmes en seront plus aisées.

La forme retenue est celle d'un langage structuré pour le contrôle programmé (branchements) et des expressions de type arithmétique pour les opérations sur les données.

L'assembleur prend totalement en charge l'implantation des variables et des micro-instructions et résout tous les cas de figures liés aux contraintes du séquenceur.

Les ressources machines apparentes ont été réduites au strict minimum, ce sont :

- Les variables "Données" et "Coefficients",
- Les pointeurs d'adresse Donnée et Coefficients,
- L'accumulateur de l'opérateur arithmétique,
- Le bus externe en anneau,
- Les diverses ressources nécessaires aux dialogues extérieurs (calculateur hôte et synchronisation).

Le programme est organisé en blocs étiquetés. Un bloc peut abriter d'autres blocs de type PROCEDURE. Les variables sont déclarées en tête avec les attributs scalaire ou tableau, réel ou complexe. Elles peuvent avoir une portée locale ou globale suivant qu'elles sont déclarées à l'intérieur ou à l'extérieur d'un bloc.

Les expressions sont de la forme :

$$\text{ACCU:} = \text{ACCU} + \text{SIGNAL} [\text{S}] * \text{COPIE} [\text{K}]$$



Les structures de contrôle sont classiques :

```

- REPEAT
  :
  :
  UNTIL <test>
- WHILE <test>
  :
  :
  END-WHILE
- IF <test> THEN
  :
  :
  ELSE
  :
  :
  END-IF
- CASE <registre> OF
  1 : ...
  2 : ...
  :
  :
  OTHERWISE ...
  END-CASE

```

Toutes ces structures peuvent être imbriquées sans exception. La structure CASE OF est plus spécialement affectée aux éclatements programmés par l'intermédiaire du registre d'éclatement du séquenceur.

AUTRES LOGICIELS

Un utilitaire éditeur de lien permet de constituer le programme final à partir de modules élémentaires écrits séparément. Cet utilitaire est également chargé de produire les fichiers de claquage des PROMs du séquenceur (PROM microprogramme, automate d'adresse, coefficients).

Le programme peut être testé et mis partiellement au point sur simulateur. Il a été créé deux simulateurs à vocation différente. Le premier décrit finement l'architecture de la machine. Il permet d'éditer tous les états logiques des différents noeuds du séquenceur et d'un opérateur. Il est particulièrement utile pour la mise au point de la machine ou étudier une phase de fonctionnement particulière.

Le deuxième simulateur est plus spécialement destiné à la mise au point des programmes opérationnels. Ce simulateur ne met en jeu que les ressources programme en exécutant le code objet produit par l'assembleur. Il est suffisamment rapide pour simuler un

traitement réel et permet d'observer en mode conversationnel toutes les ressources connues du programmeur.

En mise au point sur machine on dispose d'utilitaires logiciels implantés dans le calculateur hôte qui permettent essentiellement d'effectuer le chargement et le Dump des mémoires opérateurs ainsi que des moyens de laboratoire sous forme d'analyseurs logiques pour vérifier le déroulement des séquences programme et adresses Données et Coefficients.

CONCLUSION

On a vu qu'il était possible de réaliser avec un minimum de moyens un processeur programmable pouvant rivaliser avec les circuits câblés pour le traitement des signaux d'entrée d'un récepteur sonar actif.

Une charge de calcul voisine de 40 Mop/s correspondant à un cas concret est absorbée avec une configuration de six cartes opérateurs.

Les circuits du séquenceur essentiellement constitués de mémoires mortes et registres sont particulièrement simples à mettre en oeuvre.

La production des microprogrammes d'application est supportée par un logiciel de base conçu pour minimiser les coûts de développement grâce au langage et aux moyens de simulation mis en place.

Bien que la machine ait été conçue pour traiter les signaux de réception de sonar actif classique, son champ d'application peut être étendu à d'autres matériels dans la mesure où le traitement de plusieurs canaux identiques en parallèle peut apporter une solution au problème posé.