



HAL
open science

Étude et développement de l'informatisation du projet MALT (Mesure automatique de largeur de traits des circuits intégrés)

Annick Genin

► **To cite this version:**

Annick Genin. Étude et développement de l'informatisation du projet MALT (Mesure automatique de largeur de traits des circuits intégrés). Architectures Matérielles [cs.AR]. 1986. dumas-00322924

HAL Id: dumas-00322924

<https://dumas.ccsd.cnrs.fr/dumas-00322924>

Submitted on 19 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS-

**CENTRE AGREE
DE GRENOBLE (C.U.E.F.A)**

=====
MEMOIRE

présenté en vue d'obtenir

Le diplôme d'ingénieur

en

INFORMATIQUE

par

Annick GENIN

=====
Etude et développement de l'informatisation du projet

MALT

(Mesure Automatique de Largeur de Traits des circuits intégrés)

=====
SOUTENU LE :

JURY

Président : Mr J.Y.RANCHIN

Membres : Mr L.BOLLIET

Mr J.L.BUEVOZ

Mr M.GUILLAUME

à mon maître ,
le J et le G

REMERCIEMENTS

Le travail présenté dans ce mémoire a été réalisé dans le département Automatisation, Machines et Implantation de la division Recherche, Physique et Technologies du CNET-CNS à Grenoble.

J'exprime ma respectueuse gratitude à Monsieur J.Y RANCHIN, professeur au Conservatoire National des Arts et Métiers, qui me fait l'honneur de présider cette commission d'examen.

Je remercie vivement Monsieur Louis BOLLIET, professeur à l'Université de Grenoble 2, pour l'intérêt qu'il a manifesté au cours de ce projet.

A Jean-Louis BUEVOZ, chef du département AMI, va ma reconnaissance pour son accueil chaleureux et pour les moyens mis à ma disposition au cours de la réalisation de cette étude.

Que Michel GUILLAUME, qui a accepté d'encadrer ce projet, accepte mes remerciements pour ses précieux conseils et son amical soutien.

Je remercie Nicole ALCOUFFE-NOAILLY, ingénieur de la société MICRO CONTROLE, pour ses conseils éclairés.

Par leur gentillesse et leur patience Michel,
Louis, Jean-Claude, Gabriel et les autres ingénieurs et
techniciens du CNET ont contribué à rendre cette expérience
enrichissante.

A tous vont mes sincères remerciements.

RESUME

Ce mémoire décrit l'étude et la réalisation de l'informatisation de la méthode MALT (Mesure Automatique de Largeur de Traits).

Cette méthode est spécialement adaptée à la mesure des motifs sur plaquette dans la gamme de 0,7 à 5 microns.

MALT effectue la transformée de Fourier de la répartition d'intensité dans l'image du motif fournie par un microscope optique. Un traitement numérique tenant compte des phénomènes de diffraction détermine, à partir des zéros du spectre de l'image, la largeur du trait. La mesure ainsi obtenue est précise, reproductible, indépendante de la composition du motif mécanique et des paramètres expérimentaux. Sa justesse est meilleure que 2%.

Une première réalisation sur ordinateur (HP1000) prouve la faisabilité de MALT et le logiciel optimisé en permet l'exécution dans des temps acceptables (aux environs d'une seconde).

Une maquette de laboratoire (JAG) conçue autour d'un microprocesseur de traitement de signal (TMS 32020) montre que l'application MALT est réalisable sur une configuration informatique miniaturisée tout en conservant ses qualités de fiabilité et de justesse.

SOMMAIRE

<u>INTRODUCTION</u>	4
I <u>PREMIERE PARTIE</u> : Une approche préliminaire de MALT	7
I.1 LES CIRCUITS INTEGRES	8
I.1.1 Le transistor	8
I.1.2 Caractéristiques électriques et dimensions géométriques	9
I.1.3 Fabrication des circuits intégrés	10
I.1.4 Les contrôles	11
I.2 SYSTEMES DE METROLOGIE	12
I.2.1 Qualités demandées à un système de métrologie	12
I.2.2 Systèmes existants	13
I.2.3 Systèmes utilisant un microscope optique et un capteur à réponse linéaire	15
I.3 LA METHODE MALT	17
I.3.1 Principe de MALT	17
I.3.2 Modèle mathématique	18
I.4 REALISATION DE MALT	22
I.4.1 Dispositif expérimental	22
I.4.2 Mise en forme du signal	23

I.5	TRAITEMENT NUMERIQUE	27
I.5.1	Algorithme de MALT	28
I.5.1.1	Première estimation de la largeur du motif	31
I.5.1.2	Calcul de la FFT du signal	32
I.5.1.3	Recherche des zéros du spectre	33
I.5.1.4	Calcul de la fonction de contrôle tests de validité des résultats	36
I.5.2	Optimisations	40
I.5.2.1	Economies d'espace mémoire	41
I.5.2.2	Economies de temps	42
I.5.3	Conclusion de la première partie	48
II	<u>DEUXIEME PARTIE</u> : Une architecture micro-informatique pour MALT	49
II.1	CHOIX D'UN MICROPROCESSEUR	50
II.1.1	Choix de configuration	50
II.1.2	Qualités demandées à un processeur de traitement de signal	51
II.1.3	Processeurs en plusieurs boîtiers	53
II.1.4	Processeurs monolithiques	54
II.1.5	Notre choix	55
II.2	REALISATION DE MALT SUR TMS 32020	58
II.2.1	Description du TMS 32020	58
II.2.2	Une carte pour MALT : JAG	62
II.2.2.1	Mémoire de JAG	62

II.2.2.2 Communications avec l'extérieur	65
II.3 LOGICIEL MALT SUR TMS 32020	68
II.3.1 Système de développement	68
II.3.2 Système d'exploitation	69
II.3.3 Les structures de données	70
II.3.4 Traitement de la FFT	73
II.3.5 Traitement de la fonction de contrôle	76
<u>CONCLUSION</u>	79
BIBLIOGRAPHIE	82
ANNEXES	88

INTRODUCTION

差之毫厘
謬以千里

INTRODUCTION

En microélectronique, les circuits doivent respecter des spécifications électriques très précises. Celles-ci dépendent de la dimension des motifs élémentaires. Aussi, les contrôles systématiques des dimensions critiques, tout au long de la production, ont-ils une grande importance.

C'est dans cet objectif qu'a été mise au point la méthode MALT.

Cette méthode de mesure, dont la gamme d'application correspond aux besoins des prochaines années en métrologie (0,7 à 5 microns), effectue la transformée de Fourier de la répartition d'intensité dans l'image et recherche les minima du spectre. Un traitement mathématique, tenant compte des phénomènes de diffraction, permet d'obtenir une mesure de largeur juste, reproductible et indépendante de la composition mécanique des motifs. Le calcul d'une fonction de contrôle dont la forme qualifie la mesure, rend ce système de métrologie particulièrement fiable.

La description de cette méthode et sa réalisation informatisée, constitueront la première partie de cet exposé.

La configuration informatique choisie pour valider MALT (sur HP 1000), répond aux critères nécessaires au confort du concepteur : rapidité de calcul, mémoire vaste, programmabilité aisée mais a le gros inconvénient d'être lourde, encombrante et donc fixe. Les récents progrès de la microinformatique et l'apparition sur le marché de nouveaux processeurs de calcul rapide, nous ont incités à croire possible la réalisation de MALT sur l'un d'eux.

La deuxième partie de ce mémoire aura pour objet l'étude et la réalisation de l'implantation de MALT sur un matériel micro-informatique portable.

PREMIERE PARTIE

UNE APPROCHE PRELIMINAIRE DE MALT

...Nous nous trouvons donc dans la situation d'un homme qui ne pourrait considérer un objet qui l'intéresse qu'à travers des verres de lunettes dont il ignorerait absolument les propriétés optiques.

Max Planck

I.1 LES CIRCUITS INTEGRES

I.1.1 Le transistor

En électronique moderne, et en particulier pour les circuits logiques, les fonctions interrupteur et amplificateur sont réalisées par des transistors.

Les transistors M.O.S. (Métal-Oxyde-Semiconductor) sont particulièrement adaptés aux applications où une faible consommation est recherchée. La figure 1 représente un transistor de type PMOS.

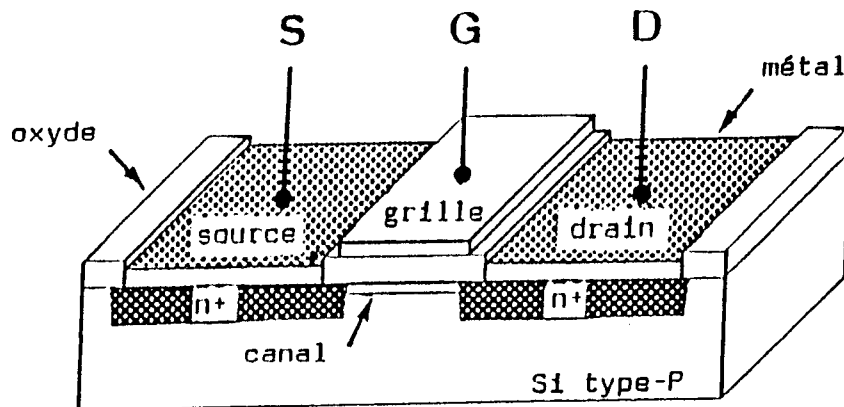


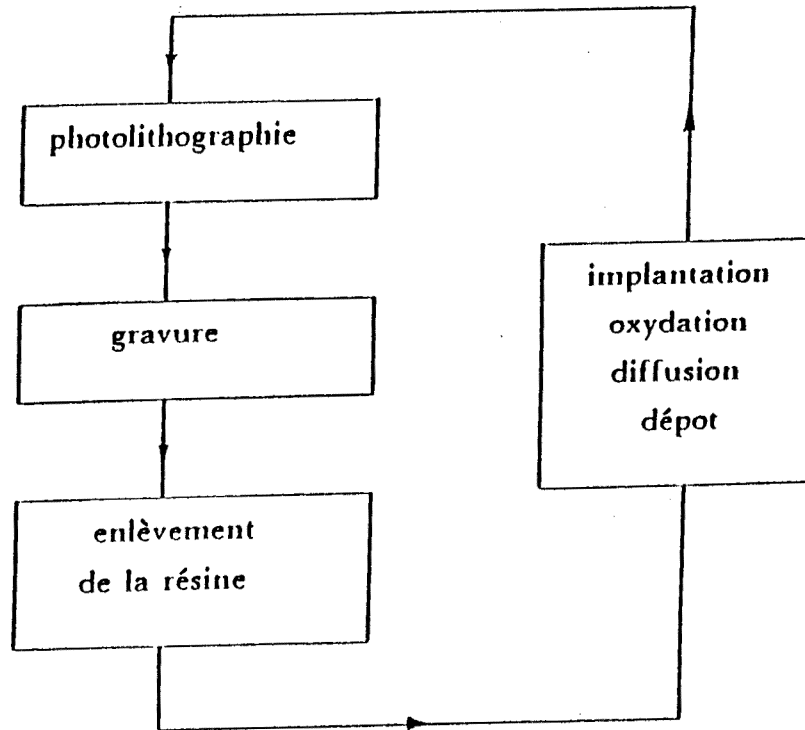
Figure 1 description des éléments formant un transistor PMOS.

Sur une plaquette de silicium, on peut fabriquer de nombreux transistors et les interconnecter pour constituer un "circuit intégré". Un tel circuit pourra effectuer des opérations plus ou moins complexes. Plusieurs centaines de circuits intégrés, ou "puces", se trouvent ainsi sur une seule plaquette de silicium (wafer). Le diamètre actuel d'une plaquette est de 100 mm et tendra rapidement vers 150 mm. En fin de fabrication, les puces sont testées, découpées, séparées puis montées dans un boîtier.

I.1.2 Caractéristiques électriques et dimensions géométriques

Les dimensions géométriques du transistor et les caractéristiques des matériaux qui le constituent permettent d'en calculer les caractéristiques électriques essentielles : tension de seuil, tenue en tension, facteur d'amplification, temps de commutation. Les performances électriques du circuit sont donc étroitement liées, d'une part à la qualité des matériaux constituant les éléments, d'autre part aux dimensions effectivement obtenues. On conçoit qu'une variation de ces dernières se répercutera immédiatement sur le comportement électrique du circuit. Celui-ci peut se voir mis hors-spécification et sa fonction électrique peut même être inutilisable.

1.1.3 Fabrication des circuits intégrés



Description d'un cycle de fabrication.

C'est l'étape de gravure qui définit la géométrie des motifs. La figure 2 montre les deux types de gravure utilisés.

La gravure humide se fait par attaque chimique et la gravure sèche par plasma.

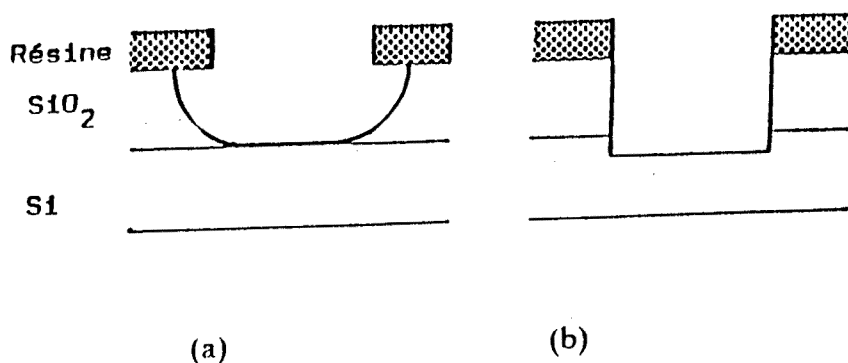


Figure 2 Profils de motifs obtenus après gravure humide (a), ou gravure sèche (b).

1.1.4 Les contrôles

Les circuits doivent respecter des spécifications électriques très précises. En fin de cycle de production, des tests fonctionnels et paramétriques permettront de détecter les circuits défectueux. Néanmoins, le cycle de fabrication durant 1 à 2 mois, des contrôles réguliers doivent être effectués afin de détecter le plus rapidement possible une éventuelle dérive des procédés.

Actuellement, nous disposons de moyens très performants pour contrôler les propriétés physiques des matériaux et les épaisseurs des couches (ellipsométrie).

En revanche, les méthodes de mesure de longueurs et de largeurs n'ont pas atteint le même niveau de qualité et sont entachées d'erreurs systématiques difficiles à estimer car dépendantes de la composition du motif à mesurer.

I.2 SYSTEMES DE METROLOGIE

I.2.1 Qualités demandées à un système de métrologie

Une mesure doit être :

- Précise : La précision est relative à une mesure isolée elle dépend de la capacité de l'appareil de mesures à détecter de faibles variations de la grandeur mesurée.

- Reproductible : La reproductibilité se caractérise par l'écart-type de la distribution des résultats. Sa valeur dépend de la sensibilité de la mesure aux phénomènes extérieurs (vibrations, températures, fatigue de l'opérateur...).

Une bonne reproductibilité se situe aux environs de 0.02 micron pour des traits de 1 à 2 microns.

- Juste : L'écart entre la valeur "admise" et la valeur "vraie" doit être minimum.

Un écart de 0.03 micron pour les traits de 1 à 2 microns correspond à une mesure "juste".

- Non destructive.

- Stable dans le temps.

- Insensible aux variations d'épaisseur des couches.

- Indépendante de l'opérateur.

- Indépendante de la nature des matériaux formant le motif.

- Parfaitement définie : Il est important de savoir si le résultat donné par la mesure correspond au sommet ou au pied du motif.

I.2.2 Systèmes existants

La plupart des systèmes de mesure dimensionnelle travaillent sur une image agrandie du motif à mesurer. Cette image est délivrée par un microscope optique ou électronique auquel est associé un équipement spécialement conçu pour assurer la fonction mesure.

Il existe 4 catégories de systèmes de métrologie dimensionnelle :

- Les systèmes utilisant un microscope électronique à balayage (MEB). L'image obtenue provient des électrons secondaires émis lors du balayage de l'objet par un faisceau d'électrons.

Le MEB permet une étude qualitative et quantitative. Grâce à son grand pouvoir de résolution, le MEB

donne accès à la topographie du motif mécanique. Les mesures que l'on peut obtenir ont une bonne reproductibilité et une justesse d'environ 2%. Le MEB permet donc de mesurer des motifs très petits (jusqu'à 0.3 micron). Les principaux défauts de cet appareil sont : sa complexité d'utilisation (mesure sous vide, calibrations fréquentes et soignées), la longueur des mesures et la destructivité (clivage et métallisation du motif). Cet instrument est donc mal adapté aux contrôles quotidiens.

- Les systèmes utilisant un microscope optique spécialement équipé pour la fonction métrologie. Cet équipement peut être :

- . Un micromètre oculaire,
- . Un oculaire à réticule,
- . Un système à dédoublement d'image,
- . Une caméra mesurant la répartition d'intensité dans l'image vidéo,
- . Un système d'acquisition linéaire de type densitométrique mesurant la répartition d'intensité dans l'image optique.

Nous détaillerons cette méthode dans le paragraphe suivant.

- Les mesures électriques sur des motifs spécifiques.

- Les systèmes qui, à l'aide d'autres instruments

optiques, effectuent un filtrage spatial ou la transformation de Fourier du trait.

I.2.3 Systèmes utilisant un microscope optique et un capteur à réponse linéaire

La correspondance entre la transition observée dans l'image et la position du bord mécanique a été étudiée par le NBS (National Bureau of Standards USA). Depuis est utilisée la notion de **seuil** qui systématise la localisation du bord de trait. Il s'agit du niveau d'intensité pour lequel la largeur du signal a une valeur proche de la largeur du trait.

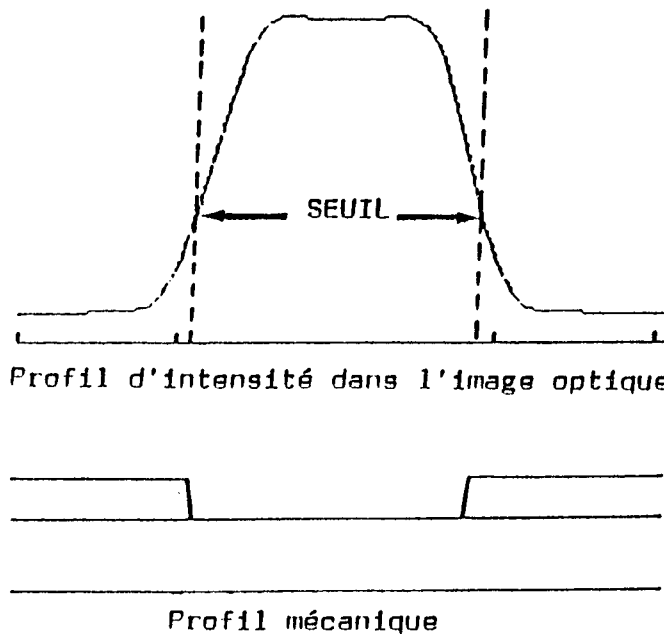


Figure 3 Correspondance entre l'image optique d'un trait et sa description mécanique.

Le NBS a montré que le seuil théorique, donnant une mesure juste, doit être calculé en fonction des paramètres optiques du motif.

Pratiquement, il est impossible de connaître les caractéristiques optiques exactes du motif et, par conséquent, le seuil théorique. Les utilisateurs effectuent donc un étalonnage pour chaque configuration de matériaux.

Les systèmes utilisant le principe d'un seuil appliqué à l'image du motif fournie par un microscope optique sont faciles d'emploi. Les mesures ainsi obtenues sont reproductibles et peu dépendantes de la mise au point pour les motifs peu épais. En revanche, ils nécessitent d'effectuer une correction qui dépend des matériaux composant le motif mécanique et aussi de la largeur du trait. Il est donc difficile d'évaluer la justesse de tels systèmes. Cette incertitude n'est pas critique pour des motifs supérieurs à 4 microns mais le devient pour des motifs voisins du micron.

I.3 LA METHODE MALT

La réduction des dimensions des circuits intégrés a rendu les appareils de métrologie insuffisants. Cette constatation a amené une équipe de chercheurs du C.N.E.T. à reconsidérer les méthodes classiques de mesure de dimension critique et à développer une technique nouvelle.

Tenant compte de la souplesse d'utilisation qu'offrent les systèmes optiques, les choix se sont portés vers un microscope optique mais en traitant les informations contenues dans l'image du trait sans utiliser le principe d'un seuil.

Ces travaux se sont concrétisés par la méthode MALT (Mesure Automatique de Largeur de Traits).

I.3.1 Principe de MALT

Si la formation de l'image d'un objet par un microscope était régie par une loi de convolution, la méthodologie de mesure de la largeur d'un motif pourrait s'inspirer des techniques de filtrage linéaire. Ainsi, en appliquant la transformation de Fourier à la fonction objet (fonction "porte" décrivant un trait idéal), celle-ci deviendrait, dans le domaine fréquentiel, de la forme sinc/x . La position des zéros de cette fonction permettrait alors de déterminer avec précision la largeur du trait.

En réalité, la théorie du microscope montre que la relation entre l'objet et son image est plus complexe qu'une simple convolution. Le modèle de formation d'image confirme cependant que la position des zéros de la transformée de Fourier de la répartition d'intensité contient l'information relative à la largeur du trait.

C'est la formulation de l'expression du spectre de l'image, tenant compte de la cohérence partielle de l'éclairage du microscope, qui conduit à la méthode de résolution MALT.

I.3.2 Modèle mathématique

La théorie de la formation d'une image en éclairage partiellement cohérent fait appel à un formalisme classique. Elle décrit l'image comme résultant de l'interaction de l'objet avec le microscope. L'action du microscope est décrite par une fonction de transfert bilinéaire dans laquelle les effets conjugués de l'éclairage et de l'objectif sont regroupés. L'objet que l'on mesure est supposé rectangulaire et caractérisé par les paramètres T_0 (réflectance relative du trait et du substrat) et F_0 (saut de phase au bord du motif). Les travaux effectués au C.N.E.T. montrent que le spectre $I(X)$ de l'image peut être exprimé comme une combinaison linéaire de deux fonctions simples $F(X,L)$ et $G(X,L)$ qui dépendent de la fréquence spatiale X , de la largeur du trait L et du facteur de cohérence de l'éclairage. Les coefficients A et B de cette combinaison

linéaire dépendent seulement des paramètres optiques du matériau : T_0 et F_0 .

$$I(X) = A * F(X, L) + B * G(X, L)$$

$$A = T_0 - 1$$

$$B = 1 + T_0 - 2\sqrt{T_0} * \cos F_0$$

$$F(X, L) = L * T(X, 0) * \sin(\pi * X * L) / (\pi * X * L)$$

$$G(X, L) = -P(X) * \cos(\pi * X * L)$$

où $T(X, 0)$ est le coefficient d'influence croisée.

Si X_i et X_j sont deux valeurs de la fréquence telles que $I(X_i) = I(X_j) = 0$, nous obtenons :

$$\frac{F(X_i, L)}{G(X_i, L)} = \frac{F(X_j, L)}{G(X_j, L)} = -\frac{B}{A}$$

Cette nouvelle relation fait disparaître l'influence des matériaux. Nous construisons alors une fonction $S_{ij}(l)$ qui s'annule pour $l = L$.

$$S_{ij}(l) = \frac{F(X_i, l)}{G(X_i, l)} - \frac{F(X_j, l)}{G(X_j, l)}$$

Ce qui, après un calcul mathématique, nous donne :

$$S_{ij}(l) = \sin(\pi(X_i + X_j)l/N) - C_{ij} \sin(\pi(X_i - X_j)N_0/N)$$

où X_i et X_j sont deux zéros quelconques du spectre et C_{ij} une fonction de la forme :

$$C_{ij} = \frac{P(X_i)X_iT(X_i)}{P(X_j)X_jT(X_j)}$$

La recherche de la largeur N_0 revient alors à

déterminer le passage à zéro de la fonction $S_{ij}(n)$.

Comme chaque paire (X_i, X_j) de zéros disponibles dans le spectre satisfait à cette relation, on utilise une nouvelle fonction $S(l)$, appelée **fonction de contrôle** qui passe par une valeur minimale pour $l = L$.

$$S(l) = \sum_{\substack{i, j=1 \\ i < j}}^M |S_{ij}(l)|$$

Où M est le nombre des zéros disponibles dans $I(k)$.

En effet, toutes les fonctions $S_{ij}(l)$ passent théoriquement par zéro pour $l = L$. Pratiquement, le signal $i(x)$ peut être distordu soit à cause de la forme du trait qui n'est pas un rectangle parfait, soit à cause d'un défaut de mise au point ou même à cause d'un mauvais réglage du microscope. Dans ce cas, les différentes paires de zéros du spectre occupent des positions légèrement différentes de ce que prévoit la théorie et les fonctions $S_{ij}(l)$ ne passent plus par zéro pour la même valeur de l . Au lieu de passer par zéro pour $l = L$, la fonction somme $S(l)$ présente un minimum dont la profondeur est d'autant plus grande que l'écart entre le modèle et l'observation est faible. Cette propriété permet d'associer un **critère de confiance** à chaque résultat de mesure. Cette information est exploitée par le système de deux manières :

- Toute mesure pour laquelle le facteur de confiance est inférieur à 90% est refusée. Ce phénomène se

produit, par exemple lorsqu'une perturbation mécanique (vibration) a affecté l'image au moment de l'acquisition. Le système annonce alors que la mesure n'est pas valable.

- Un dispositif de mise au point automatique, asservi sur la fonction de contrôle, recherche la position de la platine, à l'aide de pousseurs piézoélectriques, pour laquelle le facteur de confiance est maximal : c'est l'**autofocus**. Le système optimise ainsi lui-même les conditions expérimentales de façon à garantir une meilleure qualité du résultat.

La gamme de mesure s'étend de 0,7 à 5 microns. Au-delà de 5 microns, toute méthode de mesure à seuil est satisfaisante. La limite inférieure vient de ce que, pour que la mesure soit possible, il faut disposer d'au moins deux zéros dans le spectre; or le nombre de zéros diminue en même temps que la largeur du trait.

MALT fournit une mesure précise, reproductible, non destructive et stable dans le temps. Il fait l'objet d'un étalonnage de grandissement unique qui remplace les multiples calibrations des méthodes à seuil. La mesure est insensible aux variations d'épaisseur des couches et indépendante de la nature des matériaux.

I.4 REALISATION DE MALT

I.4.1 Dispositif expérimental

Ce dispositif a été conçu pour vérifier expérimentalement la faisabilité de la méthode décrite précédemment.

L'équipement comporte : un microscope optique, un capteur linéaire à transfert de charges, un module électronique de contrôle et un ordinateur.

* L'image optique du trait est obtenue à l'aide d'un microscope optique classique, travaillant en réflexion et disposant d'un objectif x 100.

* Le capteur CCD garantit une très bonne stabilité dimensionnelle. Il est placé dans le plan de l'image intermédiaire ce qui évite les distorsions qui seraient introduites par une optique de reprise. L'opérateur intervient pour positionner la fenêtre de mesure qui extrait l'image du trait de l'ensemble du champ observé.

* Le signal électrique délivré par chaque pixel est numérisé sur 8 bits par un convertisseur analogique/numérique intégré au module de contrôle.

* Le ordinateur HP1000 effectue la transformée de Fourier discrète rapide (sur 1024 points) du signal, recherche la position des zéros du spectre et détermine la

valeur de l qui minimise la fonction de contrôle $S(l)$. Il teste la valeur du facteur de confiance et réagit éventuellement sur la position verticale de la platine du microscope (pousseurs piézoélectriques) pour corriger la mise au point.

I.4.2 Mise en forme du signal

Cette mise en forme du signal a pour but de prétraiter le signal délivré par la barrette d'éléments photosensibles avant de le transmettre au calculateur.

Un convertisseur analogique-numérique (ADC) numérise pixel après pixel, le signal électrique délivré par la barrette. Un codage sur 8 bits permet de différencier 256 niveaux d'éclairement.

Le calcul d'une FFT (transformée de Fourier discrète rapide) ne peut être correct que si deux conditions sont respectées :

- Le motif doit être isolé.

L'opérateur définit une fenêtre spatiale contenant le trait à mesurer et rejetant les autres motifs. Il indique pour cela au système le premier et le dernier pixel décrivant le motif; ainsi le signal transmis au calculateur est constitué par l'information présente dans la fenêtre, complétée à 1024 points par une valeur constante correspondant au niveau du premier point sélectionné.

- Il ne peut y avoir de discontinuité dans le

signal.

Une différence de niveau entre le début et la fin du signal peut être interprétée comme l'addition d'une fonction "rampe" au signal origine. Si cette rampe est importante par rapport à la dynamique du signal, la FFT du signal résultant sera très perturbée en basses fréquences et la mesure effectuée ne sera pas correcte. c'est pourquoi il faut ajuster au mieux le début et la fin du signal de manière à avoir le minimum d'écart de niveaux.

Les opérations d'ajustement sont facilitées par la visualisation, en temps réel, sur un oscilloscope du signal transmis au calculateur.

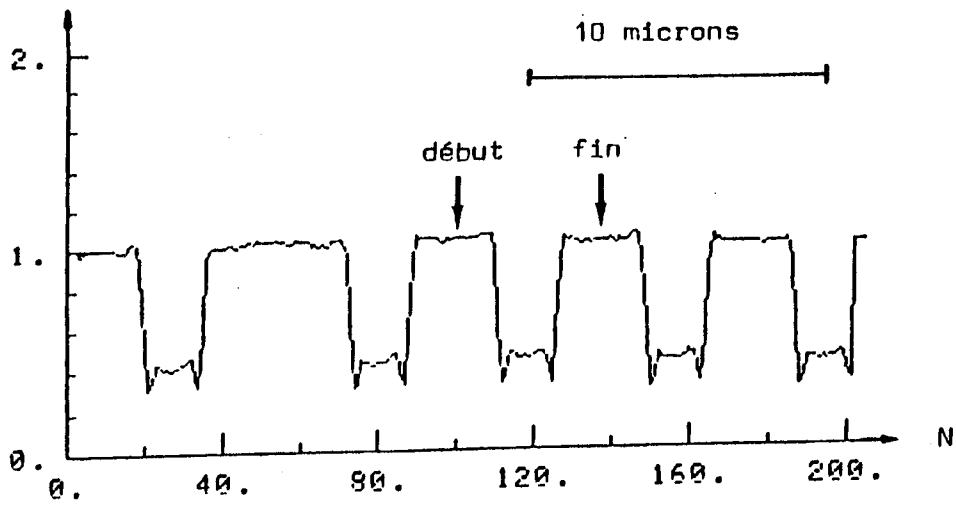


Figure 4 Signal visualisé sur l'oscilloscope.

Détermination de la fenêtre spatiale
isolant le motif à mesurer.

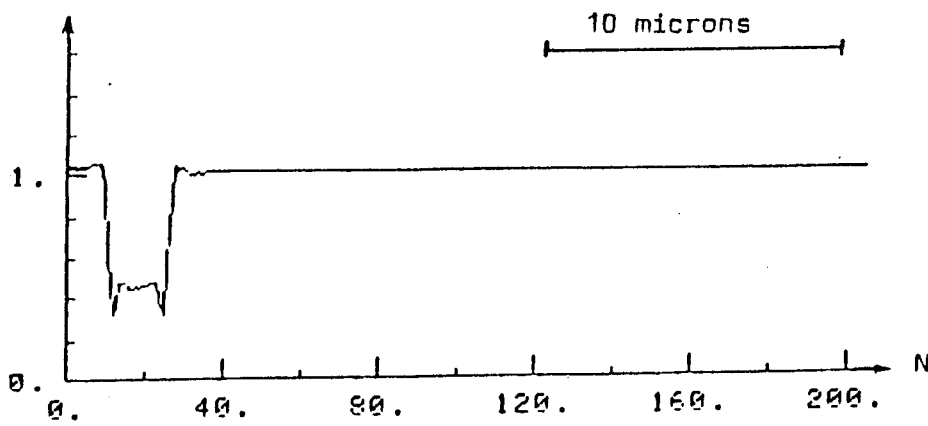


Figure 5 Signal transmis au ordinateur.

FONCTION

REALISATION

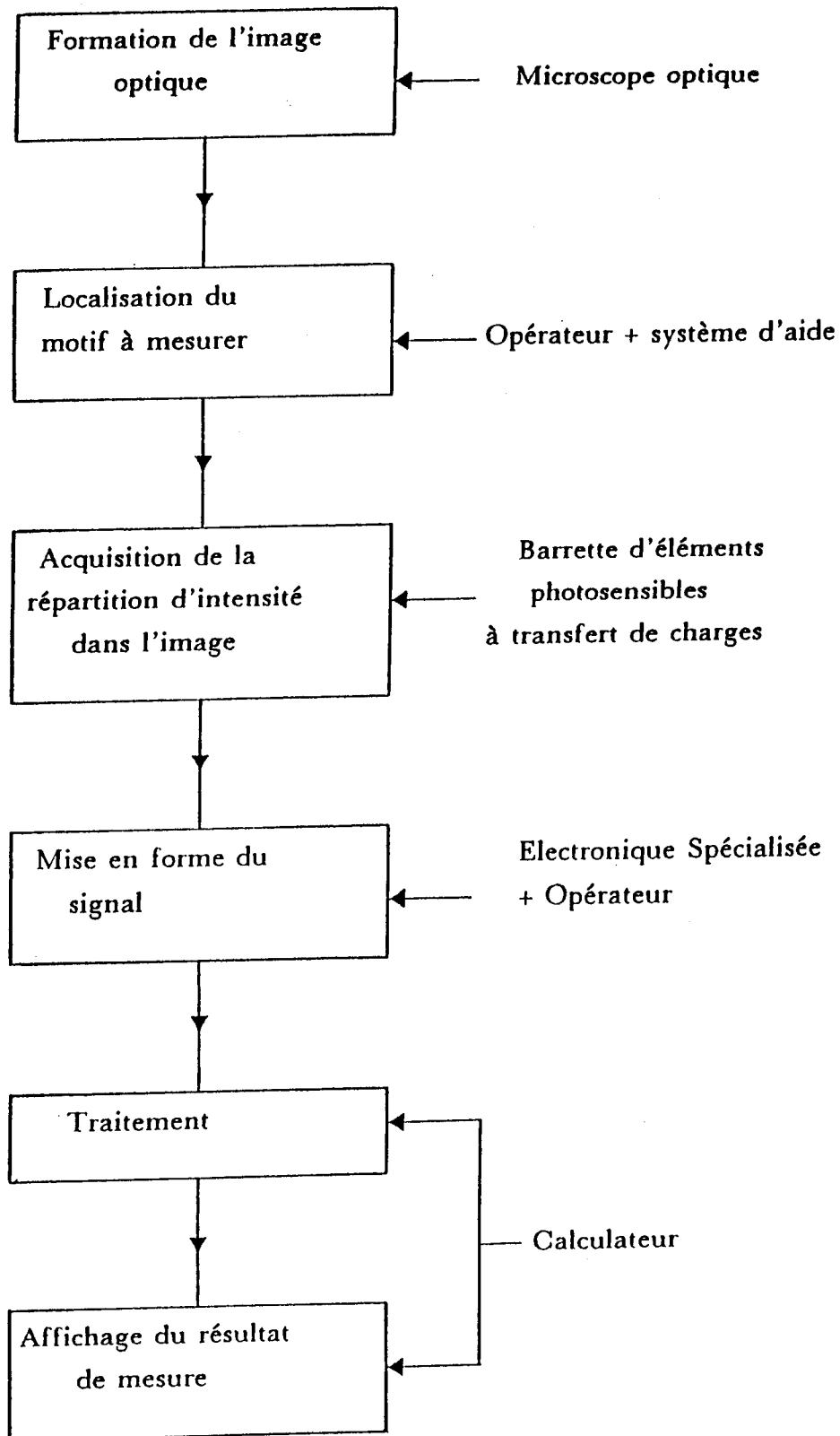


figure 6 Fonctions et organes de réalisation de MALT

I.5 TRAITEMENT NUMERIQUE

Pour valider la méthode MALT, nous disposons d'un HP1000. Ce calculateur permet le pilotage complet de l'expérience, l'acquisition et le traitement complet des signaux spatiaux, ainsi que leur modélisation. Ces signaux spatiaux peuvent, éventuellement, être stockés dans une bibliothèque en vue d'un traitement différé.

Le calculateur détermine la largeur du motif à partir du signal délivré par le capteur et prétraité par l'électronique spécialisée; c'est lui aussi qui, pour l'autofocus, commande le réglage des pousseurs piézoélectriques. Le langage choisi par les concepteurs de MALT pour mettre en oeuvre cette application, est le Fortran IV.

I.5.1 Algorithme de MALT

début

initialisations;

options : caractéristiques du matériel;

cycle

choix traitement

: traitement mesure simple;

: traitement autofocus;

: trace du signal spatial;

: trace du spectre;

: trace de la fonction de

contrôle;

: calcul du seuil;

: sauvegarde;

: options : caractéristiques

du matériel;

: fin cycle;

fin choix

fin cycle

fin

traitement mesure simple

début

acquisition du signal spatial;

première estimation de la largeur du motif;

calcul de la FFT du signal;

calcul du spectre;

recherche des zéros du spectre;

calcul de la fonction de contrôle;

test de validité des résultats;

calcul du trait et affichage des résultats;

fin

traitement autofocus

début

traitement mesure simple;

pour i=1 jusqu'à 24 faire

déplacement piézo;

acquisition du signal spatial;

calcul de la FFT du signal spatial;

recherche des zéros du spectre;

calcul simplifié de la fonction de

contrôle;

test de validité des résultats;

calcul du trait et affichage des résultats;

si meilleure mesure alors

mémoriser le signal spatial;

fin si

changer la consigne piézo;

fin faire

rappeler la meilleure mesure;

traitement mesure simple;

fin

calcul du seuil

début

acquisition du signal spatial;

première estimation de la largeur du motif;

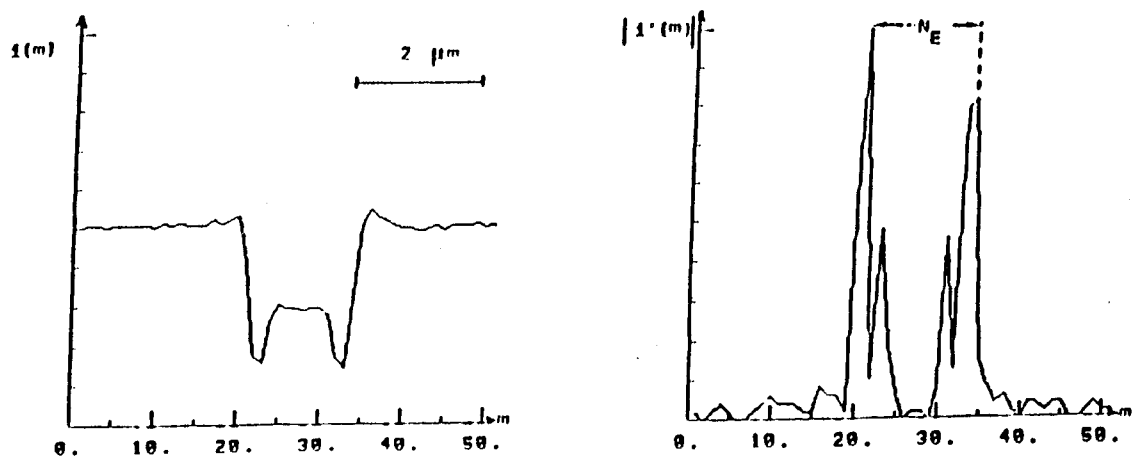
calcul du trait à partir de la première
estimation;

affichage du résultat de la mesure;

fin

I.5.1.1 Première estimation de la largeur du motif

Une estimation rapide N_E de la largeur du motif fournit une localisation de l'intervalle dans lequel il faut rechercher la solution de la fonction de contrôle. Cette estimation s'effectue à partir du signal spatial. Elle est donnée par la largeur de la répartition d'intensité dans l'image aux points de plus forte pente (dérivée maximale). Nous avons illustré ce procédé pour un motif présentant un contraste moyen.



Signal

Dérivée

Figure 7 Estimation de la largeur approximative du motif à partir de la dérivée du signal pour un trait de largeur nominale de 1,5 micron.

L'erreur relative commise par rapport à la largeur réelle du trait est d'autant plus faible que le trait est large (méthode à seuil). Cette erreur devient négligeable pour une largeur de trait supérieure à 5 microns.

I.5.1.2 Calcul de la FFT du signal

L'algorithme le plus classique de calcul de transformée de Fourier numérique est celui de COOLEY-TUKEY.

Il s'applique à tous les signaux complexes (ou réels) dont le nombre de points d'entrée est une puissance de 2. Cet algorithme, s'il ne nécessite pas un grand espace mémoire, a un temps d'exécution relativement long (plus de 2 secondes sur HP1000 en Fortran pour 1024 points complexes).

Le signal d'entrée étant réel, il existe une optimisation de cet algorithme qui conserve la simplicité de sa mise en oeuvre (annexe 1).

Pour l'application sur HP1000, nous avons choisi un algorithme plus complexe, s'appliquant aux signaux réels dont le nombre de points est une puissance de 8 et de 2. L'espace mémoire demandé est grand (5 fois plus grand que pour l'algorithme précédent), mais le temps d'exécution moindre (2 fois plus petit).

Le spectre du signal est calculé par l'expression :

$$I(k) = \sqrt{I_R(k)^2 + I_J(k)^2}$$

où $I_R(k)$ est la partie réelle et $I_J(k)$ la partie imaginaire de $I(k)$.

1.5.1.3 Recherche des zéros du spectre

La recherche des zéros de $I(k)$ revient à la recherche des points de rebroussement du spectre. Du fait de l'échantillonnage fréquentiel, ces minima atteignent rarement le niveau zéro. pour calculer précisément la position de ces points, on utilise les échantillons situés de part et d'autre du minimum. La position z_m du zéro est le point d'intersection des droites A et B (figure 8).

$$z_m = \text{min} + \frac{2 \cdot I'(\text{min}+1) - I'(\text{min}+2) - 2 \cdot I'(\text{min}-1) + I'(\text{min}-2)}{I'(\text{min}-1) - I'(\text{min}-2) + I'(\text{min}+1) - I'(\text{min}+2)}$$

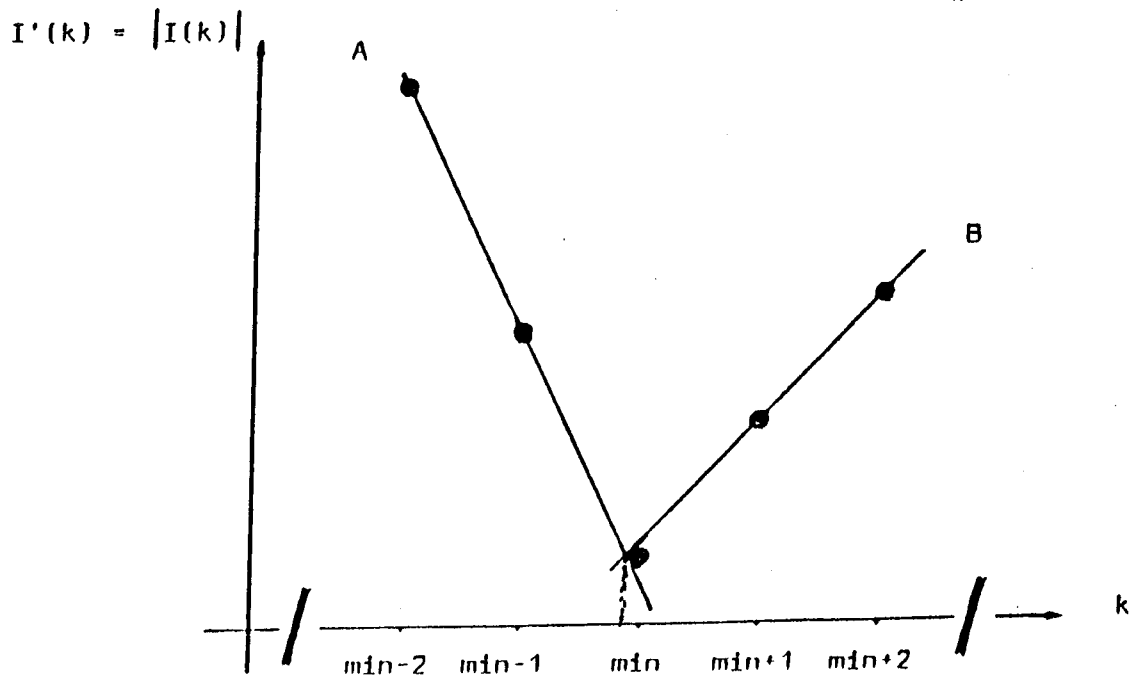


Figure 8 Détection de la position de rebroussement à partir des échantillons encadrant le minimum.

Il peut arriver que le spectre présente un premier minimum qui n'est pas un point de rebroussement. Nous testons la valeur du spectre à ce point; si $I(\min)$ est supérieur à $XMAX/2$ ($XMAX$ est l'amplitude maximale de $I(k)$), ce minimum est alors rejeté.

Pour éviter de prendre en compte des zéros trop perturbés par le bruit, par exemple lorsque l'amplitude du spectre décroît très rapidement, on limite la plage de recherche des zéros à $1,3 \cdot F_c$ (où F_c est la fréquence de coupure cohérente du système optique).

I.5.1.4 Calcul de la fonction de contrôle **Tests de validité des résultats**

L'estimation de la largeur du motif effectuée au début du traitement permet de limiter la zone d'exploration de la fonction de contrôle $S(l)$ autour de la valeur N_E obtenue après première estimation.

Soit S_{MAX} la valeur maximale de $S(l)$ sur l'intervalle ainsi défini. Le pas de calcul de n définit la précision de la mesure. Si nous prenons un pas de calcul très petit, le temps de calcul sera trop long. Pour l'expérimentation, $S(l)$ est calculée par pas de 0,1 point soit 0,013 micron. A l'approche du minimum de $S(l)$ (niveau inférieur à $S_{MAX}/10$), le pas d'échantillonnage est réduit à 0,02 points soit 0,0026 micron.

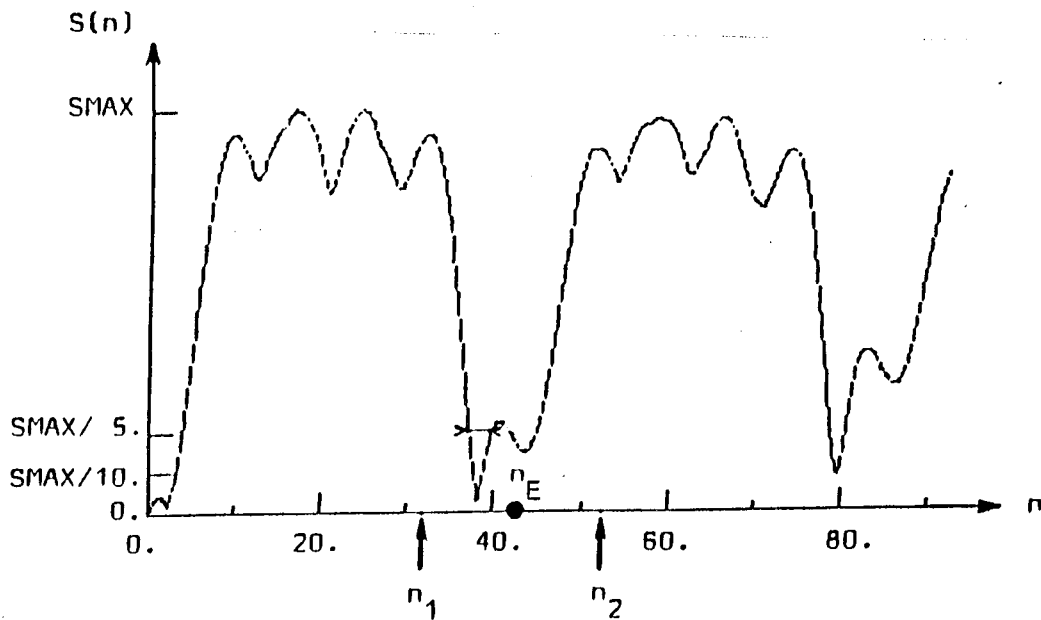


Figure 9 Exemple de fonction de contrôle $S(n)$ obtenue pour un motif de 2 microns à partir de 5 zéros présents dans le spectre.

Pour éviter les décisions erronées, sont introduits deux paramètres importants :

- La largeur du pic de $S(n)$ au niveau $S_{MAX}/5$ est-elle inférieure à 0,2 micron (4 points) ? Si c'est le cas, il faut rejeter la mesure et changer le réglage du microscope (couleur).

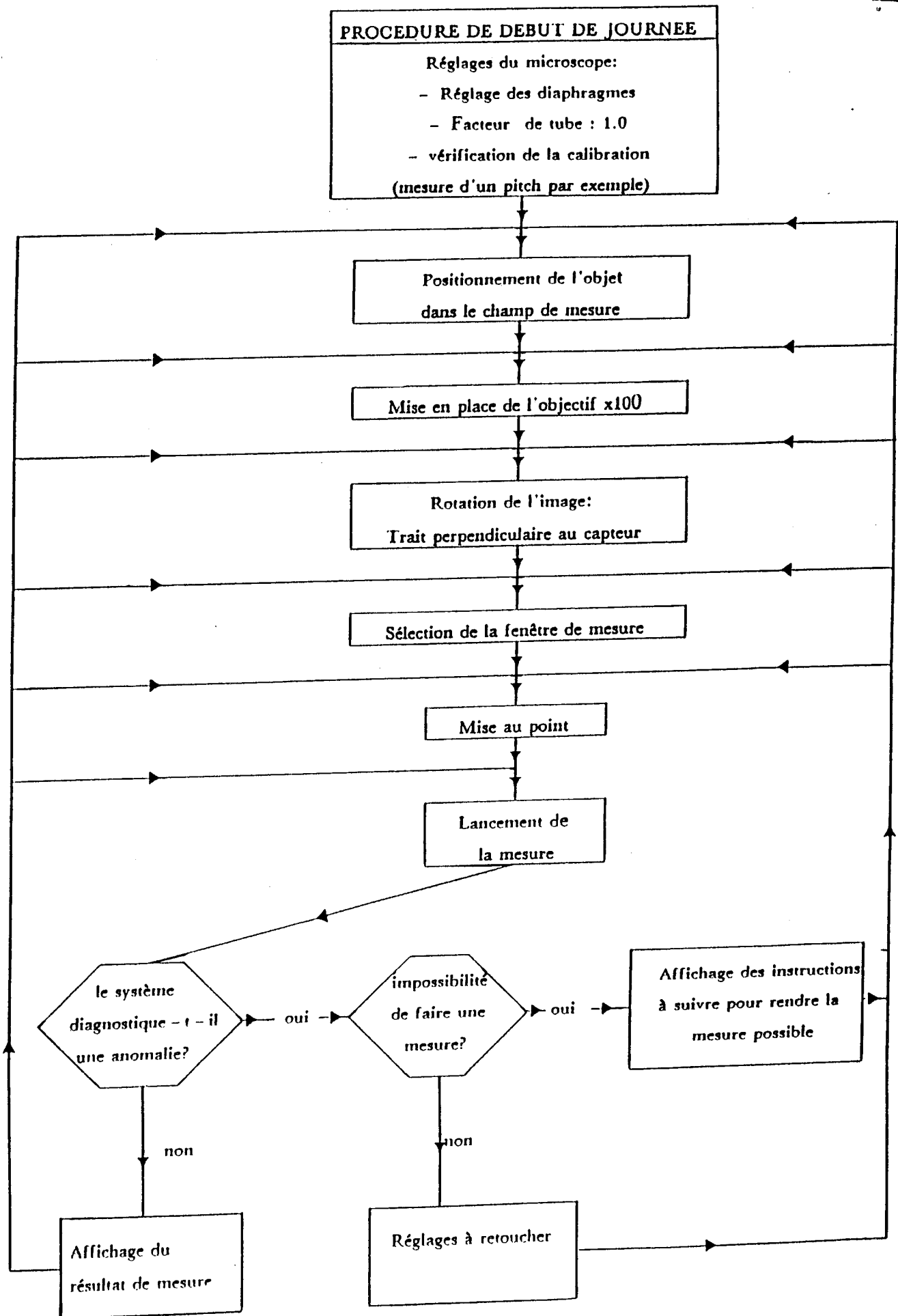
- Le pic est-il assez profond ? Si le minimum ne descend pas en dessous de $S_{MAX}/10$, il faut également rejeter la mesure, même si le pic est étroit. En effet, ce phénomène, interprété comme un défaut d'homogénéité de l'ensemble des zéros, signifie qu'il y a une mauvaise mise au point ou un

mauvais réglage des diaphragmes ou encore une anomalie dans la sélection de la fenêtre spatiale. Il a été constaté, lors des expériences de validation, que ces tests sévères évitent la plupart des erreurs d'interprétation de résultats.

Remarquons que plus le trait est petit, moins il y a de zéros dans le spectre et donc plus le calcul est rapide.

Pour les motifs larges, le nombre de zéros a été limité à 9 afin de conserver un temps de calcul raisonnable.

Pour les motifs supérieurs à 5 microns, l'utilisation de la première estimation de la largeur à partir du signal spatial fournit une mesure dont l'incertitude ne dépasse pas 5% de la valeur réelle. Ce procédé, donnant une mesure très rapide, augmente le confort de l'opérateur.



Organigramme d'une journée de mesures

I.5.2 OPTIMISATIONS

Dans un premier temps, la validation a été effectuée sans la partie autofocus. Le souci essentiel était alors d'exécuter l'ensemble des fonctions de calcul, c'est-à-dire le traitement de la mesure simple, dans un temps "acceptable" pour un mode interactif.

Le temps d'attente jugé acceptable pour un opérateur moyennement patient est de l'ordre de la seconde pour une mesure et ne doit, en aucun cas, dépasser 4 secondes.

L'espace mémoire était largement suffisant puisque nous disposons de 30K mots (de 16 bits) de mémoire centrale sur le HP1000.

Avec l'apparition de l'autofocus, se sont posés des problèmes de deux natures :

1- L'espace en mémoire centrale s'est avéré insuffisant.

2- Le temps de calcul, qui sans l'autofocus pouvait dépasser 1 seconde, a paru très long lorsqu'il a été multiplié par 25.

C'est afin de remédier à ces carences que nous avons entrepris l'optimisation de MALT. La remise en cause est purement informatique et ne doit à aucun moment modifier

la sémantique de l'application.

I.5.2.1 Economies d'espace mémoire

L'espace mémoire est occupé par les données et les programmes.

* L'espace données

Puisque l'espace n'était pas compté, les données étaient stockées au fur et à mesure des besoins. En particulier, des espaces de travail pour les données temporaires étaient créés pour chaque utilisation différente.

La structure séquentielle de l'algorithme autorise la réutilisation d'un même espace de travail pour les diverses procédures. Nous avons donc défini un espace de travail commun à toutes les procédures.

D'autre part, les tableaux avaient été très largement dimensionnés de manière à pouvoir modifier les algorithmes sans se soucier de leur taille. L'algorithme défini, il a été possible de réduire considérablement les bornes des tableaux. Par exemple, le signal d'entrée est acquis sur 1024 points alors que seules les 100 premières valeurs sont significatives.

Le passage à une arithmétique entière ou à virgule fixe, a pour conséquence une réduction de l'espace mémoire. En effet, les nombres réels (flottants simples) sont codés sur 32 bits alors que les entiers, comme les nombres à

virgule fixe, ont une longueur de 16 bits. Le passage du format réel au format entier induit donc une réduction de la taille de la mémoire de moitié pour les variables ainsi transformées.

Cette transformation peut entraîner une perte de précision (en format flottant, la mantisse est codée sur 24 bits). Toutes les mesures effectuées en arithmétique entière et à virgule fixe délivrent une largeur de trait égale à celle calculée en valeurs réelles ou ne différant pas de plus de 0,01 micron.

* L'espace programme

L'espace occupé par le code a pu être réduit en regroupant des séquences dupliquées au cours du programme. Cependant, certains calculs sont reproduits plusieurs fois avec les mêmes valeurs car leur regroupement obligerait à stocker les résultats dans des tableaux ; or le temps d'accès à ces tableaux est plus important que celui pris en réeffectuant les calculs.

L'espace mémoire utile à l'application MALT a été ramené à 22K mots (de 16 bits).

I.5.2.2 Economies de temps

C'est l'objectif prioritaire dès que l'espace mémoire est suffisant.

Dans le déroulement d'un programme, ce sont les

procédures contenant beaucoup de calculs et les communications avec l'extérieur qui sont longues à exécuter.

Le programme MALT comporte :

- Un dialogue opérateur qui consiste en l'acquisition des choix de traitement au début de chaque mesure et éventuellement, en l'affichage des courbes demandées (signal spatial, spectre, fonction de contrôle). Ces fonctions sont visualisées sur l'écran de la console et ne réclament pas une très grande rapidité, les ralentissements étant dûs essentiellement aux temps d'accès aux périphériques et à la vivacité de l'opérateur.

- Deux fonctions de calcul : la FFT et le calcul de la fonction de contrôle qui doivent demeurer transparentes à l'opérateur et donc s'exécuter très rapidement. Ce sont ces deux fonctions que nous chercherons à optimiser.

*** Calcul de la FFT**

Pour calculer une FFT sur N échantillons, il est nécessaire d'effectuer $(N-1)^2$ multiplications et $N(N-1)$ additions complexes. Plus N augmente, plus le temps de calcul et l'occupation mémoire s'accroissent. C'est pourquoi des algorithmes ont été définis permettant de réduire le nombre de ces opérations par une bonne organisation des calculs.

Nous avons conservé pour le calcul de la FFT un algorithme très performant au point de vue temps mais très gourmand en mémoire (2096 mots pour la FFT d'un signal sur 1024 points réels en bases 8 et 2 contre 545 mots pour le même algorithme en base 2).

Nombre de points de la FFT.

Nous avons vu que la transformée de Fourier était calculée sur 1024 points. L'adjonction de points ayant tous la même valeur augmente la densité des points d'information dans le domaine fréquentiel. La FFT sur 1024 points offre une bonne sécurité. Nous avons cependant observé la position des zéros pour des signaux sur 1024 points et les avons comparés aux zéros obtenus pour les mêmes signaux mais ramenés à 512 points. Nous n'avons pas remarqué de différences notables. Les petites différences constatées nous ont parues tout à fait négligeables en comparaison de l'incertitude du signal et des erreurs dues aux approximations du calcul de la FFT.

Le tableau suivant montre le temps d'exécution en secondes, sur HP1000, de la FFT d'un **signal réel** programmée en Fortran.

	base 2	bases 2 et 8
1024 points	0,70s	0,38s
512 points	0,33s	0,17s

Nous constatons que le temps de calcul peut être grandement réduit en prenant moitié moins de points pour le calcul de la FFT.

Si N est le nombre de points de la FFT, nous retrouverons la valeur des zéros sur 1024 points en multipliant ceux obtenus par $1024/N$.

Passage en arithmétique entière

La FFT supporte très bien d'être calculée en format entier. En effet, le signal d'entrée est codé sur 8 bits et, au cours des différents calculs de la FFT, seul le premier point risque d'être sujet aux débordements. Le premier point n'étant pas considéré dans notre application (c'est la composante continue qui correspond à l'éclairage moyen de l'image et ne transporte pas d'information utile), il n'est pas une entrave à la réalisation en format entier. Pour garder l'homogénéité des formats, les multiplications par les fonctions trigonométriques peuvent être traitées en virgule fixe.

D'autre part, le calcul du spectre sert à déterminer les passages à zéro. Nous ne nous intéressons qu'à ses valeurs proches de zéro. En conséquence, nous pouvons affirmer qu'il n'est pas nécessaire d'en calculer la racine carrée et que les valeurs excédant un format de 15 bits peuvent être tronquées à la valeur $2^{15}-1$. Les calculs ainsi effectués sont plus rapides et évitent les imprécisions introduites par le calcul de la racine carrée.

Remarquons que la recherche des zéros du spectre se fait sur le premier quart des points calculés. Il n'est donc pas nécessaire, dans le dernier papillon, de calculer tous les points.

* La fonction de contrôle

La procédure de calcul de la fonction de contrôle

est grande consommatrice de temps. La durée de résolution de cette fonction est directement liée au nombre de zéros. Plus le nombre de zéros (non aberrants) est grand, plus le résultat est "juste".

Nous avons réduit le calcul de la fonction de contrôle en ne considérant qu'un point sur deux, ce qui raccourcit de moitié la recherche approchée du minimum. La recherche fine examine les points ainsi écartés.

Un pas de 4 eût été optimum mais, à partir de 3, le risque de laisser échapper le minimum de la fonction de contrôle devenant important, nous avons refusé cette solution qui risquait de compromettre le résultat.

Un autre moyen de minimiser la durée du calcul, a été de réduire la plage de recherche du minimum. Pour une mesure simple, c'est la première estimation de la largeur qui centre la plage de recherche. Cette estimation étant très approximative, le calcul est effectué sur une longueur de 200 points. Dans l'autofocus, la première mesure simple délivre une largeur beaucoup plus "juste" que la valeur estimée. Nous sommes donc partis de ce dernier résultat pour centrer la plage de recherche que nous avons ramenée à 20 points.

Nous avons ainsi pu réduire de 10% le nombre de calculs dans une boucle qui pouvait contenir jusqu'à 4000 sinus, 7000 multiplications et 1000 divisions.

Nous avons vérifié que la fonction de contrôle pouvait être transposée en format virgule fixe sans dommage pour les résultats.

Il est à noter que, sur le HP1000, le temps d'exécution des calculs en format entier ou à virgule fixe, est très peu différent de celui des mêmes calculs effectués en format réel flottant.

Evaluation du temps d'exécution d'une mesure simple (FFT + recherche du minimum de la fonction de contrôle) en fonction du nombre de zéros pour une FFT sur 1024 points calculée en bases 2 et 8 (0,38s) :

nombre de zéros	temps de calcul en secondes
2	0,47
3	0,52
4	0,59
5	0,67
6	0,76
7	0,88
8	1,03
9	1,17

I.5.3 Conclusion de la première partie

Conformément à ce que prévoyait la théorie, MALT permet d'obtenir une mesure qui présente, pour des largeurs de 0,7 à 5 microns, des caractéristiques de précision (0,003 micron), de reproductibilité (écart de 0,02 micron) et de justesse. Sa réalisation sur HP1000 répond de façon satisfaisante au modèle théorique. L'optimisation des programmes a ramené l'espace mémoire et le temps d'exécution à des dimensions acceptables, tout en conservant la précision nécessaire à une bonne mesure.

DEUXIEME PARTIE

UNE ARCHITECTURE MICRO-INFORMATIQUE POUR MALT

Il faut développer de nouveaux outils de conception qui seront à la fois plus simples, plus rapides et plus sûrs.

Camus

II.1 CHOIX D'UN MICROPROCESSEUR

Dans la première partie de cet exposé, nous avons montré que le HP1000 convenait bien à l'application MALT. Cependant, la microinformatique a énormément évolué ces dernières années dans le sens de la rapidité d'exécution, de la souplesse d'utilisation et la maniabilité. Certains microprocesseurs concurrencent la vitesse de calcul des ordinateurs les plus puissants.

Aussi avons-nous pensé que l'un d'eux pourrait accueillir MALT en respectant toutes ses caractéristiques. c'est l'étude d'une telle réalisation que nous aborderons au cours de la deuxième partie de ce mémoire.

II.1.1 Choix de configuration

Le programme MALT comporte, comme nous l'avons vu, un dialogue opérateur et une partie calculs. Le dialogue opérateur est réalisable sur la plupart des micro-ordinateurs dès qu'ils offrent des possibilités graphiques, sa seule contrainte étant d'être raisonnablement rapide mais pas plus qu'un homme devant un terminal.

Lorsqu'ils sont correctement programmés, les micro-ordinateurs savent tout faire, même les calculs les plus complexes, mais ils prennent leur temps ce qui est proprement inadmissible à notre époque : en conséquence, on leur adjoint des circuits spécialisés dans le calcul

arithmétique ou des coprocesseurs mathématiques qui se font une joie de jongler à toute vitesse avec les nombres.

Nous nous orientons donc, pour réaliser le traitement informatique de MALT, vers une configuration comprenant **un micro-ordinateur standard pour gérer le dialogue opérateur et, pour remplir la fonction de calcul, un processeur rapide.**

Le choix d'un processeur rapide est délicat et dépend directement de l'application; aussi allons-nous décrire les possibilités qui nous sont offertes.

II.1.2 Qualités demandées à un processeur de calcul

- Performance : Elles sont annoncées par le constructeur. Elles décrivent le temps d'initialisation et la vitesse de calcul. Cependant, il ne faut pas se fier à la seule vitesse de calcul. Ainsi, pour les multiplieurs les plus rapides, le chargement des opérandes-sources et le prélèvement du résultat sont plus pénalisants que le temps de multiplication. Il faut aussi tenir compte des possibilités d'accès à des mémoires rapides.

Le nombre d'opérations effectuées, les formats acceptés, les contrôles et comptes-rendus sont à considérer dans les performances.

- Compacité : La comparaison ne doit pas porter sur la taille du composant de calcul seulement, mais sur l'ensemble du matériel nécessaire pour mettre en oeuvre

l'application. Il peut s'agir de logique de décodage, de l'étage de mémorisation, de logique d'interface...

- Programmabilité : Il s'agit de l'enchaînement des opérations élémentaires pour la réalisation de calculs complexes. Les mécanismes de chaînage d'opérations vont de l'accumulateur des multiplieurs au décodage direct des instructions, concurremment au CPU, comme le font les coprocesseurs.

La programmabilité s'exprime aussi par la liberté offerte à l'utilisateur d'établir ses propres algorithmes à l'intérieur même du composant : certains microprocesseurs, destinés en priorité aux applications de traitement de signal, sont tout à fait utilisables pour des applications d'arithmétique générale (contrairement à leurs concurrents directs).

- Adaptabilité : Elle consiste en la présence d'options et la facilité de réutilisation dans d'autres configurations. C'est une notion parfois essentielle puisque la volonté d'utiliser à nouveau une même unité arithmétique sur deux microprocesseurs différents pénalise fortement les circuits trop spécialisés comme les coprocesseurs.

- Disponibilité : Elle recouvre divers aspects. L'annonce d'échantillons n'implique pas la commercialisation immédiate. La disponibilité aux Etats-Unis n'est pas une preuve de disponibilité en France.

Pour clarifier une situation parfois complexe, nous avons classé les circuits arithmétiques en deux catégories : ceux dont la structure est figée, souvent monolithiques, et ceux composés de plusieurs boîtiers spécialisés.

Nous trouverons en annexe 2 une étude comparative de quelques uns de ces circuits.

II.1.3 Processeurs en plusieurs boîtiers

Avec un processeur en plusieurs boîtiers, le concepteur peut réaliser une maquette dont les chemins de transfert seront optimisés pour l'application donnée. Il devra réaliser une étude de matériel longue et coûteuse.

Les composants bien choisis seront regroupés sur une ou plusieurs cartes selon une architecture adaptée à l'application.

Dans le cas de MALT, il serait nécessaire de câbler deux cartes : l'une spécialisée dans le calcul de la FFT, en ayant déterminé son nombre de points en entrée; l'autre adaptée aux calculs arithmétiques et trigonométriques.

Le tout serait commandé par un microprocesseur et disposerait d'une mémoire commune à l'ensemble afin de minimiser les échanges externes.

Il existe, sur le marché, des cartes conçues pour être enfichées directement sur des micro-ordinateurs courants. Ces cartes ont de très bonnes performances mais sont grandement ralenties par les accès mémoire.

Ce type de réalisation n'est vraiment très performant que s'il est spécifique à l'application.

II.1.4 Processeurs monolithiques

Un processeur à architecture figée minimise le temps d'étude mais rend plus critique le choix du processeur à installer.

On peut les diviser en deux catégories : les processeurs indépendants (pouvant être programmés de manière autonome), et celle des opérateurs dédiés, appelés coprocesseurs, optimisés pour le couplage à un microprocesseur donné.

* **Les processeurs autonomes.** Nous ne considérerons que les processeurs de traitement de signal (les autres étant trop lents pour notre application). Ils sont de deux sortes : ceux qui intègrent la multiplication, et les autres. La deuxième catégorie ne couvre qu'un nombre restreint d'applications, et pour l'étude d'un processeur de traitement de signal à usage général, il est obligatoire de s'occuper du mode de réalisation de la multiplication.

Parmi les processeurs de traitement de signal, celui qui semble le mieux correspondre à nos critères de rapidité, de programmabilité et de facilité de mise en oeuvre est le TMS 32020 de Texas Instruments.

* **Les processeurs dédiés,** ou coprocesseurs, datent des microprocesseurs 16 bits. Le premier représentant fut le

8087 coprocesseur du 8086 ou 8088 d'Intel. Motorola propose le 68881 associé au 68010 ou 68020, National Semiconductor le 16081 et Zilog le 8070. Ces coprocesseurs s'accomodent de plusieurs formats d'opérandes et de multiples fonctions arithmétiques, exponentielles et logarithmiques.

Celui qui présente les meilleures performances de rapidité est le 68881 de Motorola.

II.1.5 Notre choix

Si nous devons résumer MALT en quelques chiffres, nous dirions :

- Une FFT qui, pour 1024 points, comprend :
 - 18.944 additions et soustractions,
 - 11.264 multiplications.
- Un ensemble de calculs qui peut atteindre :
 - 18.000 additions et soustractions,
 - 4.500 divisions,
 - 13.000 multiplications,
 - 9.000 sinus.

Le tout devant être exécuté en moins d'une seconde.

Ces opérations, nous l'avons vu, peuvent être réalisées soit en format flottant, soit en format entier et à virgule fixe.

L'ensemble de MALT peut occuper jusqu'à 22 K mots de 16 bits de mémoire.

Pour étayer notre choix, nous avons réalisé une

étude comparative des différents circuits disponibles dits de "calcul rapide". Nous avons tenu compte de leurs performances (annexe 2), mais aussi de leur facilité de mise en oeuvre.

Nos contraintes étaient :

- éviter toute limitation sur les algorithmes pouvant être réalisés;
- exclure toute règle de programmation inhabituelle ou exotique;
- obtenir une réalisation peu coûteuse et donc utiliser des microprocesseurs universels.

Les fonctions de calculs étant assez diverses, nous avons écarté la solution des boîtiers multiprocesseurs. En effet outre la complexité de conception, une solution de ce type figerait l'application et empêcherait des évolutions ultérieures.

Restaient donc les processeurs monolithiques. Parmi ceux-ci nous avons retenu soit le TMS 32020, le plus agréable des processeurs de traitement de signal bien que n'étant pas le plus rapide, soit le coprocesseur 68881, le plus rapide des coprocesseurs, associé au 68020.

Le second, séduisant par ses nombreuses fonctions et sa grande programmabilité, est cependant plus lent que le TMS 32020, son environnement de développement plus coûteux. D'autre part, des difficultés d'approvisionnement de ce produit ont orienté notre choix vers le **TMS 32020**.

Ce microprocesseur offre des performances très convenables pour MALT. Texas Instrument sortira bientôt des versions encore plus performantes de ce circuit (version CMOS

plus rapide et offrant un jeu d'instructions plus grand). De nombreux constructeurs développent actuellement des cartes et logiciels à base de TMS 32020 ce qui laisse espérer la sortie prochaine de compilateurs de langages de haut niveau (C, Pascal, Fortran).

II.2 REALISATION DE MALT SUR TMS 32020

II.2.1 Description du TMS 32020

Nous allons décrire brièvement l'architecture et le jeu d'instructions de ce processeur de traitement numérique du signal (DSP (Digital Signal Processing)), "VLSI" (Very Large Scale Integration).

* Architecture

Le bloc diagramme du processeur, ainsi que ses principales caractéristiques, apparaît en annexe 3.

A l'intérieur, le processeur est organisé en une structure double bus programme/données. Les blocs de mémoire RAM intégrés sont dynamiquement reconfigurables soit comme des espaces "données" contigus soit comme des espaces "données" et "programme" séparés.

Extérieurement, le circuit n'utilise qu'un seul bus pour communiquer avec des mémoires "programme" ou "données" et avec des entrées/sorties. Ainsi, des programmes contenus dans les mémoires externes peuvent également être exécutés à vitesse maximale en utilisant l'ensemble des 544 mots de mémoire RAM comme mémoire de données. Le processeur permet également que les espaces programme et données soient situés à l'extérieur du circuit.

L'adressage des opérandes est effectué en mode d'adressage direct ou indirect. L'adressage direct est

spécifié par l'adresse d'un mot à l'intérieur d'une page de 128 mots dans le code même de l'instruction. Un pointeur de page de données composé de 9 bits indique le reste des 16 bits d'adresse de manière à déterminer laquelle des 512 pages doit être appelée pour l'instruction en cours.

L'adressage indirect utilise l'un des 5 registres auxiliaires devant servir de pointeur pour les opérations mettant en jeu la mémoire. Une unité arithmétique séparée, et spécifique aux registres auxiliaires, permet au registre auxiliaire d'être incrémenté, décrémenté ou indexé suivant le contenu d'un autre registre auxiliaire, durant le même cycle. Ce dernier mode est souvent utile quand il est nécessaire de balayer une zone de données par incrémentation non unitaire comme, par exemple, dans le calcul d'une transformée de Fourier rapide.

Des structures et instructions spéciales ont été incorporées afin de permettre une gestion efficace des espaces mémoires internes et externes. Par l'utilisation de l'entrée "READY", des mémoires et circuits périphériques lents peuvent être utilisés par le biais de la génération d'un état d'attente. Les transferts de blocs de données sont facilités par la possibilité de transférer d'une zone de mémoire à une autre de grands blocs de programme ou de données. Quand elle est utilisée avec l'entrée "READY", cette possibilité de transfert par blocs permet à des programmes d'être chargés en mémoire RAM à partir de mémoire externe plus lente comme une EPROM (Erasable Programmable Read Only Memory). Une autre caractéristique de ce circuit offre à une

partie de la mémoire externe la possibilité d'être configurée comme mémoire de données globale.

L'unité arithmétique et logique (ALU) de ce processeur réalise les opérations arithmétiques signées et non signées sur des mots de 16 bits avec un résultat sur 32 bits mémorisé dans l'accumulateur.

Un multiplieur 16x16 bits peut réaliser un produit sur 32 bits à chaque cycle. Est également présent, un registre permettant le décalage à gauche des opérandes avant qu'ils ne rentrent dans l'ALU. L'ALU possède deux entrées, l'une étant la sortie de l'accumulateur, tandis que l'autre vient soit du registre "produit" du multiplieur soit de la sortie du registre à décalage. Un registre à décalage à la sortie du registre "produit" rend possible l'alignement à gauche des produits fractionnaires aussi bien que le décalage à droite des produits avant qu'ils n'entrent dans l'ALU. Des décalages à droite et à gauche dans l'accumulateur sont aussi possibles.

*** Les instructions**

Les instructions de multiplication/addition en un seul cycle rendent aisées les sommes de produits. Ces instructions utilisent pleinement la bande passante du multiplieur par le passage de deux opérandes à la fois à chaque cycle. Ceci s'effectue par l'utilisation d'un registre auxiliaire pour adresser l'un des opérandes et par l'utilisation du compteur de programme pour adresser l'autre.

Les deux bus intégrés acheminent ensuite les opérandes vers les entrées du multiplieur en même temps que le produit précédent est additionné à l'accumulateur en exécution de l'opération d'addition.

Plusieurs autres instructions : arithmétiques, de contrôle, générales ont été aussi incorporées sur le circuit.

Des instructions ont été prévues afin de calculer la valeur absolue, le complément à un et à deux, aussi bien que la représentation normalisée d'un nombre de 32 bits contenu dans l'accumulateur. La division est assurée par la présence d'une instruction de soustraction conditionnelle. Certaines instructions effectuent des opérations de test au niveau du bit sur des mots de la mémoire, d'autres effectuent des opérations arithmétiques et logiques directement sur 16 bits.

Le jeu d'instructions du processeur est orienté de manière à fournir un maximum de flexibilité et d'efficacité au système. Les instructions de répétition permettent de répéter la seule instruction suivante jusqu'à 256 fois. L'utilisation de cette possibilité de répétition combinée à certaines instructions précédemment mentionnées, telles que les transferts de bloc ou la multiplication/somme, peut conduire à une suite de codes très compacte là où, autrement, il aurait fallu un long programme linéaire ou des branchements conditionnels consommateurs de temps.

II.2.2 Une carte pour MALT : JAG

Le microprocesseur, comme nous l'avons vu, doit supporter la partie calcul de MALT. Pour conserver intacte sa puissance et sa rapidité de calcul, il est indispensable de minimiser les échanges avec l'extérieur et en particulier avec un ordinateur hôte. C'est donc le processeur lui-même qui commandera les acquisitions des valeurs en provenance du microscope; il ne laissera la main au micro-ordinateur hôte que lorsque tous les calculs auront été effectués.

Le TMS 32020 est un produit récent et nous n'avons pas trouvé de carte, pouvant nous satisfaire, à base de ce processeur. Nous avons donc décidé de réaliser une carte convenant à notre application. C'est ainsi qu'est née JAG (acronyme composé des initiales de ses concepteurs). Cette carte doit réaliser entièrement la fonction calcul de MALT. Elle a donc été conçue avec un espace mémoire pouvant contenir toute l'application et des organes de communication avec l'extérieur.

II.2.2.1 Mémoire de JAG

Comme nous l'avons vu, le TMS 32020 dispose d'une mémoire RAM interne de 544x16 bits. Cet espace n'est pas suffisant aussi faut-il adjoindre des mémoires externes :

*** Pour profiter au mieux des performances du**

microprocesseur, il est indispensable de posséder des mémoires très rapides.

Nous trouvons actuellement sur le marché, à des prix très abordables, des mémoires **RAM suffisamment rapides** (<85ns) pour être accédées par le microprocesseur dans le temps d'un cycle.

Dans ces conditions, la plupart des instructions pourront être exécutées dans les mêmes conditions de temps que si elles étaient stockées en mémoire interne. Il en est de même pour les données. Cette mémoire volatile servira à ranger les données acquises et à être chargée avec les programmes à exécuter rapidement.

* **Pour fonctionner de façon autonome**, il est nécessaire d'avoir accès à de la mémoire non volatile (mémoire de masse) pouvant conserver tout le logiciel de l'application et le noyau de système servant à la mise en route. Des mémoires de type **EPROM** conviendront. Leur inconvénient est leur lenteur (au moins un temps de cycle de pénalisation pour les plus rapides).

Afin de répondre à toutes ces contraintes, nous avons choisi de mettre sur la carte des EPROM (32K mots de 16 bits), mémoire de masse, stockant l'ensemble du logiciel nécessaire à la fonction de calcul de MALT et à son démarrage. D'autre part, nous disposons de mémoire RAM très rapide (70ns), suffisamment vaste (24K mots de 16 bits) pour contenir les programmes et données.

La phase de mise en route consiste à transférer le programme de calcul dans la mémoire RAM depuis l'EPROM afin d'en avoir une exécution sans dégradation des performances du microprocesseur.

Il est à remarquer que dans le TMS 320, les mémoires peuvent être configurées soit en mémoire programme, soit en mémoire données; ce qui, éventuellement, permet à la même adresse, de faire correspondre deux mémoires (une mémoire programme et une mémoire données) les signaux DS et PS faisant la sélection.

Notre espace étant suffisant, nous avons choisi de ne pas différencier la mémoire programme de la mémoire données.

*** Répartition de l'espace mémoire.**

Nous disposons de trois groupes de mémoire RAM (RAM1, RAM2, RAM3), adressant chacun 8K mots de 16 bits soit, en tout, 24K mots et d'un groupe d'EPROMS contenant 32K mots de 16 bits. A la mise sous tension, le microprocesseur va chercher la première instruction à l'adresse 0. Il est donc essentiel que cette adresse 0 se trouve située dans la mémoire de masse. En conséquence, nous avons réservé à la mémoire EPROM les adresses les plus basses ($A_{15} = 0$). La répartition de la mémoire se fera selon les adresses suivantes :

EPROM : 0000 - 7FFF

RAM1 : 8000 - 9FFF

RAM2 : A000 - BFFF

RAM3 : C000 - DFFF

II.2.2.2 Communications avec l'extérieur

La carte fonctionne de manière autonome, mais encore faut-il lui transmettre les données acquises sur le microscope et doit-elle faire connaître les résultats de ses calculs.

Pour la transmission des données, la carte porte une interface d'entrée/sortie parallèle 16 bits, qui, outre l'acquisition des données en provenance de la barrette, peut recevoir et transmettre des informations.

Pour les dialogues avec un ordinateur hôte et une console, nous avons choisi deux liaisons standard RS232 (CCITT V24), la liaison RS232 étant la plus utilisée des interfaces pour les liaisons à courte distance. Bien que moins rapide que le mode synchrone, nous avons choisi le mode série asynchrone moins astreignant à gérer que le premier.

La carte étant bien définie, elle a été dessinée sur une machine de D.A.O. CALMA (annexe 4).

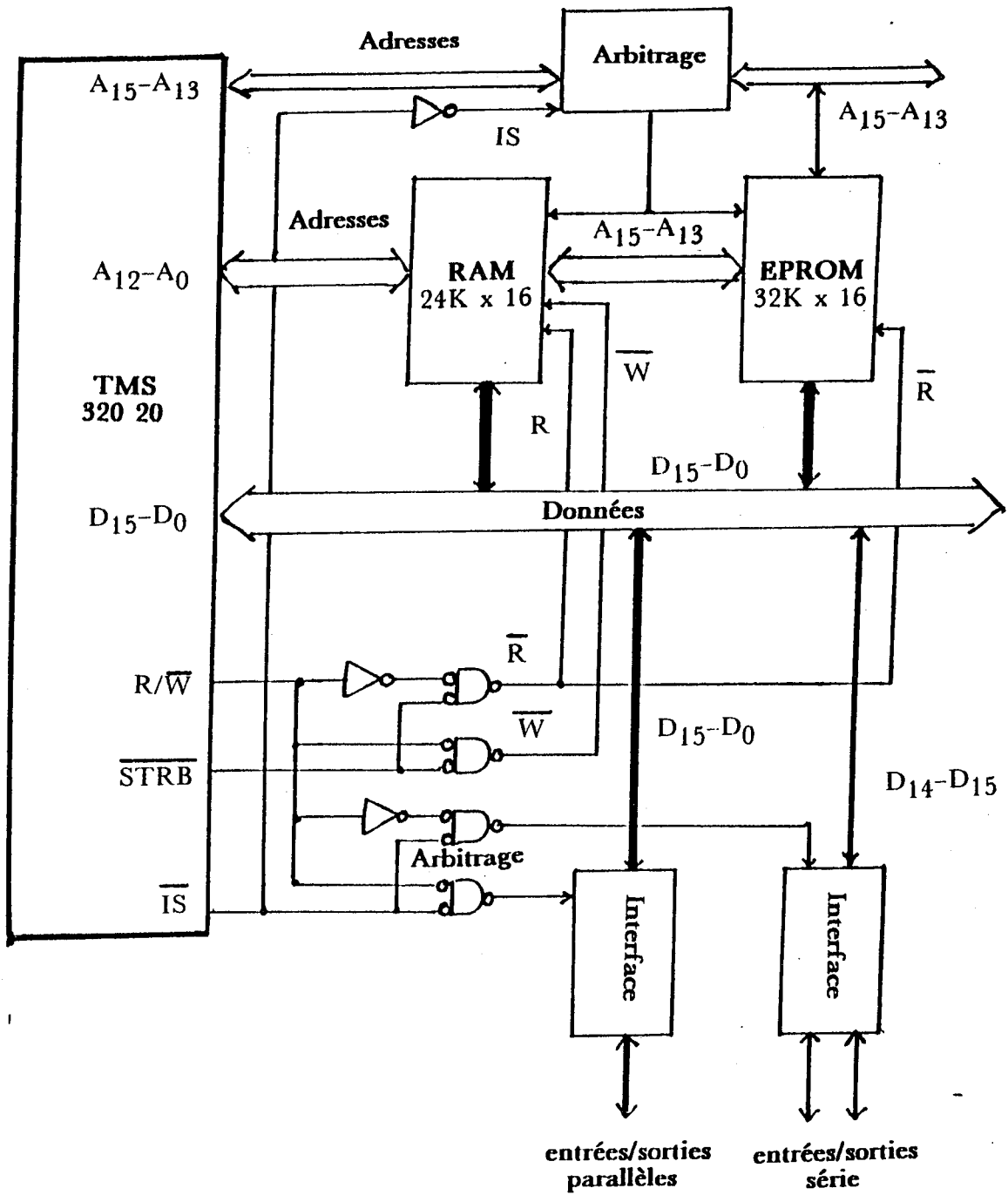
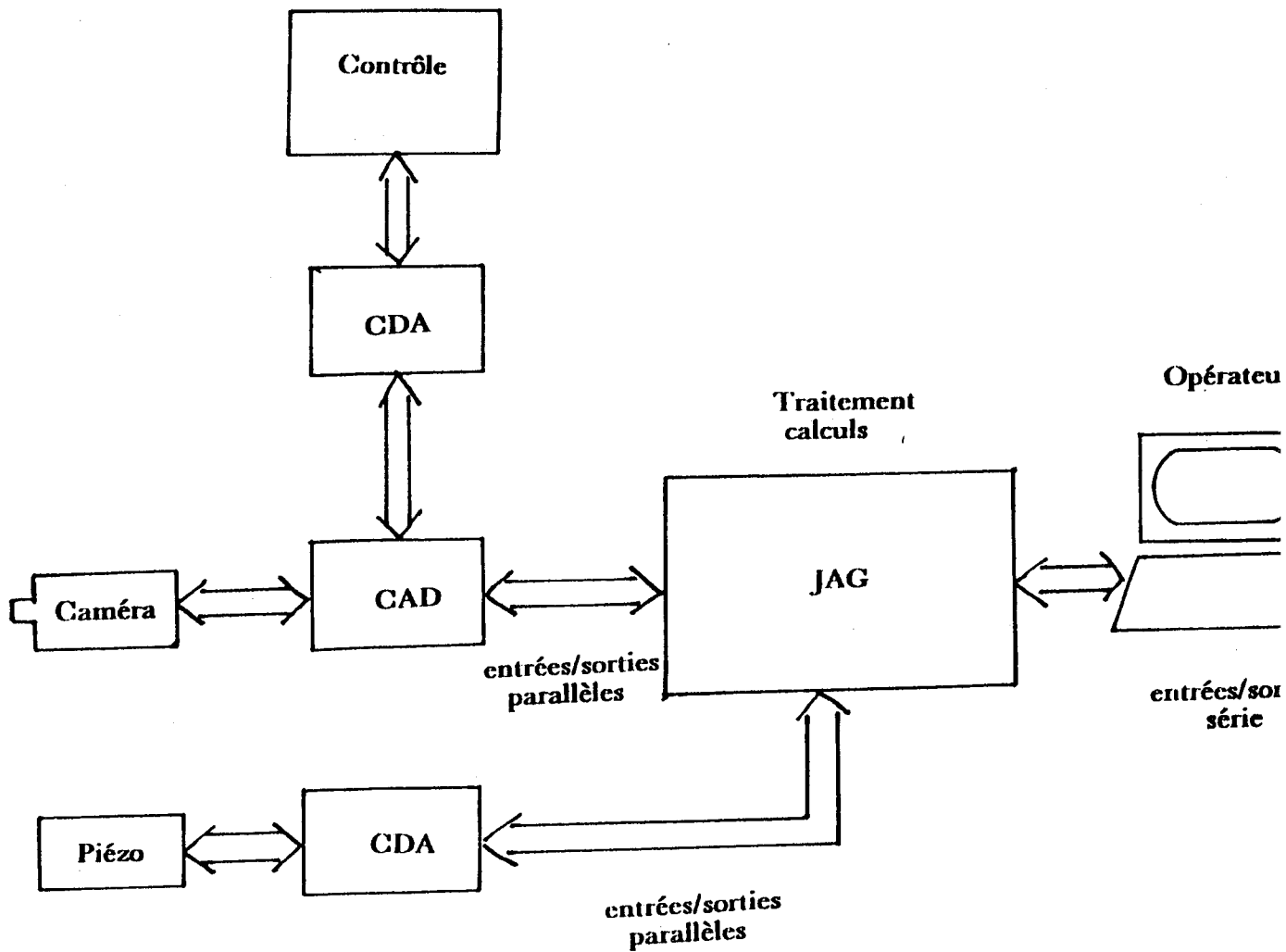


Schéma de principe de la carte JAG



Principe de la solution proposée

II.3 LOGICIEL MALT SUR TMS 32020

L'ensemble du logiciel de la carte est composé d'un noyau de système minimum et du programme MALT.

MALT comporte un algorithme de FFT, un algorithme de résolution de la fonction de contrôle pour une mesure simple, le même algorithme, simplifié, pour l'autofocus et quelques algorithmes de poursuite de minimum.

Après avoir décrit l'environnement du système de développement, nous exposerons les fonctions du système d'exploitation. Nous aborderons les options prises pour les structures de données communes à ces procédures puis nous détaillerons l'implantation de la FFT.

II.3.1 Systeme de développement

Le TMS 32020 peut être émulé par un système de développement qui lui est propre, l'XDS 20.

Pour des raisons économiques et pour ne pas avoir à acquérir du matériel qui ne serait pas utilisé en dehors de cette application, nous avons renoncé à l'achat d'un XDS 20 et nous avons décidé d'utiliser le matériel informatique disponible au C.N.E.T..

La mise au point du logiciel s'effectue donc par l'intermédiaire d'un système de développement comprenant :

- * Un VAX qui supporte l'assembleur et l'éditeur de liens du TMS 32020. Il n'existe pas, actuellement, de compilateur de langage de haut niveau.

- * Un PDP 11 MINC qui recueille le code objet en provenance du VAX et qui gère le dialogue opérateur.

- * Un SCH qui émule le microprocesseur, directement enfiché sur la carte à laquelle il tient lieu de mémoire de masse, en remplacement des EPROMS.

Les programmes sont chargés en RAM, sur la carte, à partir du SCH.

Une des sorties RS232 de la carte est reliée au PDP 11 qui gère la console opérateur.

II.3.2 Système d'exploitation

Il ne suffit pas de ranger le programme dans la mémoire pour que celui-ci s'exécute; il faut gérer un minimum de système d'exploitation.

Le programme système a pour fonctions :

- Lancer l'exécution du programme au moment de la mise en route.

- Transférer les programmes de la mémoire lente (EPROM) vers la mémoire rapide (RAM). Cette opération peut être faite facilement et rapidement grâce aux instructions de répétition.

- Gérer les entrées/sorties. C'est la partie la plus conséquente. Elle englobe la gestion des lignes RS232, respectant le format de transmission des données, la temporisation imposée pour une bonne synchronisation des émissions et réceptions. On y trouve aussi les conversions ASCII, binaires, hexadécimales.

Nous avons écarté le traitement des interruptions qui ne sont pas nécessaires à la mise au point. Elles pourront être traitées ultérieurement.

A ces fonctions, nous avons ajouté quelques utilitaires de mise au point tels que l'accès à une mémoire, l'affichage et la modification d'une mémoire, le branchement à une adresse. Ces utilitaires seront retirés de la version définitive.

II.3.3 Les structures de données

La structure des instructions du TMS 32020, intégrant les registres auxiliaires, encourage le rangement des données sous forme de tables.

Dans l'application Fortran sur HP1000, les tableaux utilisés ont été définis; nous reprenons donc des structures semblables soit :

- un tableau de 100 éléments pour l'acquisition des valeurs en provenance du microscope ;

- un tableau de 1024 éléments pour calculer la FFT et qui délivrera les valeurs utilisées du spectre ;

- des tableaux conservant les informations relatives aux zéros et aux coefficients qui leurs sont associés ;

- un espace de 1024 mots contigus servant de zone de travail et de rangement des variables temporaires.

A cet espace déjà défini, il faudra adjoindre une **table des sinus** et cosinus.

En effet, si la fonction sinus n'est ni câblée ni microprogrammée, le calcul d'un sinus ou d'un cosinus est effectué par programme à l'aide d'approximations. Cette méthode assez longue devient très pénalisante lorsque plusieurs centaines de sinus sont à calculer.

Afin de gagner du temps, les valeurs des sinus sont précalculées et rangées dans une table dont la taille dépend de la précision que l'on veut obtenir.

Nous cherchons à construire une table accessible facilement et minimisant au maximum les calculs d'indices.

Pour le calcul de la fonction de contrôle, tous les angles sont positifs et seules interviennent les fonctions

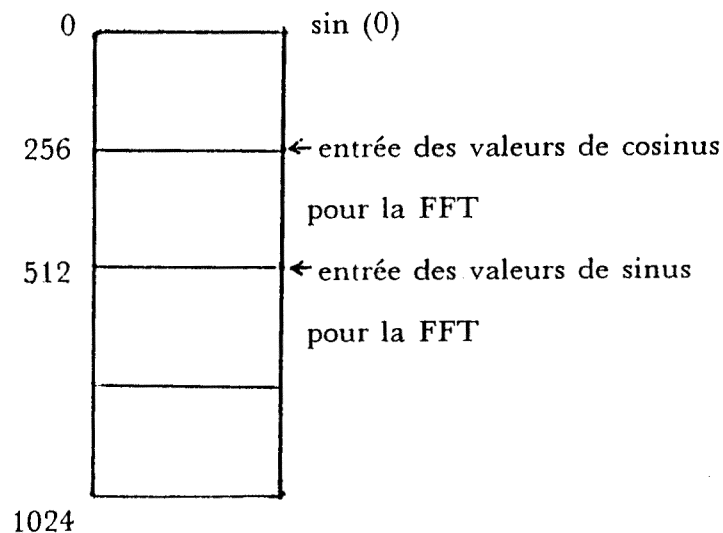
sinus.

Pour connaître l'ensemble des valeurs trigonométriques de la transformée de Fourier et de la fonction de contrôle, il suffit de connaître les valeurs de sinus d'angles compris entre 0 et 2π .

Le nombre d'entrées dans cette table est fonction de la précision que l'on veut obtenir. La transformée de Fourier sur 1024 points considère des angles dont le "pas" minimum est $\pi/512$. Ce "pas" est tout à fait suffisant pour la fonction de contrôle qui n'exige pas une très grande précision.

La table des sinus la plus satisfaisante pour l'application MALT sera donc celle contenant les valeurs de sinus compris entre 0 et 2π sur 1024 points.

Organisation de la table des sinus :



II.3.4 Traitement de la FFT

Le TMS 32020 est bien adapté aux tâches de traitement de signal. Les instructions de multiplication et d'addition (ou soustraction) combinées sont particulièrement favorables au calcul de la FFT.

Un examen (effectué par Texas Instruments pour le TMS 320) des algorithmes utilisant diverses bases a montré que les performances (temps et mémoire) atteintes avec la base 2 sont meilleures que celles de la base 4 ou 8 qui utilisent de nombreuses variables intermédiaires.

*** Algorithme de la transformée de Fourier d'un signal réel.**

L'algorithme le plus couramment utilisé est celui qui considère le signal d'entrée comme complexe (COOLEY-TUKEY). Si le signal d'entrée est réel, il suffira d'adjoindre à chaque valeur une composante imaginaire nulle pour pouvoir appliquer la théorie de la FFT complexe. Il existe cependant une méthode qui, à partir de la théorie complexe, fournit la FFT d'un signal réel avec beaucoup moins de calculs. Cette méthode n'étant pas communiquée dans tous les ouvrages de référence, nous en rappelons brièvement le principe :

Soit la suite réelle $(x_n, n=0,1,\dots,2N-1)$

$$\text{on veut calculer } X_k = \sum_{n=0}^{2N-1} x_n W_{2N,nk}$$

$$\text{où } W_{2N,nk} = \exp(-2\pi jnk/2N)$$

Les valeurs en entrée x_n étant réelles, on a la symétrie :

$$X_{2N-k} = X_k$$

Il suffit donc de connaître les valeurs $X_k, k=0,\dots,N$ pour connaître tout le spectre.

L'algorithme de la FFT complexe sur N points permet de calculer :

$$\begin{aligned} Y_k &= \sum_{n=0}^{N-1} (x_{2n} + jx_{2n+1})W_{N,nk} \\ &= \sum_{n=0}^{N-1} x_{2n}W_{N,nk} + j \sum_{n=0}^{N-1} x_{2n+1}W_{N,nk} \\ &= G_k + j H_k \end{aligned}$$

pour $k = 0,1,\dots,N-1$.

Or on veut :

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_{2n}W_{N,nk} + (\sum_{n=0}^{N-1} x_{2n+1}W_{N,nk})W_{2N,k} \\ &= G_k + W_{2N,k}H_k \end{aligned}$$

$$Y_{N-k} = G_{N-k} + jH_{N-k} = G_k + jH_k$$

donc

$$Y_{N-k} = G_k - jH_k$$

$$\text{et } Y_k = G_k + jH_k$$

entraîne

$$G_k = (Y_k + Y_{N-k}) / 2$$

$$\text{et } H_k = -j(Y_k - Y_{N-k}) / 2$$

d'où le calcul d'un dernier papillon :

$$X_k = G_k + W_{2N,k}$$

$$X_{N-k} = G_k - W_{2N,k}H_k$$

$$X_0 = G_0 + H_0$$

$$X_N = G_0 - H_0$$

$$X_{N/2} = Y_{N/2} \quad (\text{puisque } G_{N/2} \text{ et } H_{N/2} \text{ sont réels}).$$

Le calcul de la transformée de Fourier de 2N points réels revient à calculer la FFT de N points complexes en ajoutant un papillon spécifique à la transformée réelle. C'est dans ce dernier papillon qu'est calculé le spectre.

Ainsi que nous l'avons étudié, les valeurs des sinus et cosinus sont rangées dans une table à laquelle on accède directement. En effet, $\sin(2\pi nk/2N)$ est le i ème élément de la table des sinus avec $i = nk \cdot 1024/2N$, la table contenant 1024 éléments.

Contrôle des débordements

Dans le cas de MALT, le contrôle des débordements n'intervient pas. Le signal d'entrée est codé sur 8 bits et seul le premier point risque de dépasser le format 16 bits or ce premier point ne nous intéresse pas.

Il n'y a donc pas lieu de procéder à des recadrages.

Erreurs de calcul

La principale source d'erreurs, à part les

imprécisions dues aux approximations, est celle introduite par l'arrondi après multiplication des sinus (ou cosinus). Cette différence n'influe pas sur la position des zéros, en conséquence, nous ne nous en inquièterons pas.

II.3.5 Traitement de la fonction de contrôle

L'implémentation de l'algorithme de la fonction de contrôle sur TMS 32020, présente deux difficultés :

- la représentation des nombres à virgule ;
- les nombreuses divisions.

Représentation des nombres à virgule :

Le TMS 32020 supporte les nombres flottants codés sur 32 bits mais au prix d'un temps assez long. D'autre part, ces nombres flottants ont une mantisse codée sur 16 bits ce qui ne donne pas une meilleure précision que pour les nombres à virgule fixe.

Les nombres à virgule sont tous du même ordre de grandeur, il est donc facile de gérer la virgule et de recadrer tout au long des calculs.

Nous garderons le format à virgule fixe.

Division des nombres :

L'instruction de division n'existe pas sur le TMS 32020. Pour les divisions par des puissances de deux, il suffit d'utiliser des décalages à droite ou de passer dans la

partie haute (16 bits de poids forts) de l'accumulateur par décalage du nombre d'éléments binaires à conserver.

Pour les divisions décimales, nous utilisons l'algorithme suivant, en tenant compte du fait que nous divisons toujours des nombres positifs (si ce n'était pas le cas, il faudrait, en plus, gérer le signe).

Le jeu d'instructions inclut une soustraction conditionnelle très utile à la division. Cette instruction effectue la soustraction si le dividende est plus grand que le diviseur sinon, le dividende est multiplié par 2 (décalage à gauche).

Algorithme de la division de deux nombres positifs à virgule :

début

n = 15 + nombre d'éléments binaires après

la virgule du quotient ;

compteur de répétition = n ;

charger le numérateur dans l'accumulateur ;

répéter n fois

soustraction conditionnelle du

dénominateur ;

fin répéter

fin

Cette division utilise n+2 cycles machine (de 200ns chacun) ce qui, pour une précision de 5 éléments binaires

après la virgule, représente $200\text{ns} * 20 = 4000\text{ns}$ soit $4\mu\text{s}$.

Ce temps de calcul d'une division est tout à fait acceptable dans notre application.

CONCLUSION

不
长
一
智

不
经
一
事

CONCLUSION

La méthode MALT que nous venons de décrire a été réalisée sous forme de maquette d'étude dans les laboratoires du C.N.E.T.. Le traitement numérique est assuré par un HP1000. Dans la gamme de 0,7 à 5 microns, la mesure obtenue est précise, reproductible et juste.

L'optimisation des programmes, sans entamer les qualités de MALT, a amélioré les conditions de rapidité d'exécution et agrandi la disponibilité de mémoire permettant de mettre en place un système d'autofocus et laissant la possibilité d'adjoindre d'autres fonctions.

Le temps n'étant plus une limite, il devient possible de complexifier les algorithmes et d'effectuer des vérifications plus poussées sans en craindre la lenteur.

L'étude et l'état de réalisation actuelle de MALT sur un microprocesseur de traitement de signal montrent la possibilité d'implantation de MALT sur un système miniaturisé, facilement transportable sur les lieux d'expérimentation.

Nous avons opté pour la séparation du traitement du

dialogue opérateur, réalisable sur la plupart des micro-ordinateurs actuels, et du traitement des calculs exigeant un processeur rapide.

La maquette de mise au point ne constitue pas un prototype industriel dans la mesure où certaines fonctions, dans un souci de simplicité de mise en oeuvre, ont été réduites à leur minimum. C'est ainsi que les entrées/sorties ont été traitées de façon simple n'utilisant pas toutes les possibilités de synchronisation du TMS 32020. Il serait, en particulier, intéressant d'établir une liaison rapide entre la carte et le micro-ordinateur hôte qui gèrera le dialogue opérateur et les fonctions graphiques.

Nous avons conçu une carte de calcul rapide, JAG, possédant une mémoire vaste et très rapide. Cette carte fonctionne de manière autonome et est capable de délivrer les résultats des calculs sans intervention extérieure.

Le logiciel supporté par la carte est susceptible d'évoluer et d'être amplifié sans avoir à modifier l'architecture de l'ensemble.

Le matériel non informatique nécessaire au fonctionnement de MALT constituera la partie "lourde" et encombrante tandis que le système se chargeant du traitement numérique sera "léger" et transportable.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

M.CAND, E.DEMOULIN, J.-L.LARDY & P.SENN

Conception des circuits intégrés MOS.

EYROLLES, 1986

J.J.CHRISHOLM

A new technique for linewidth measurement on 1 micron geometry process wafers using fluorescence.

SPIE Proc, Vol 480, pp 49-56, mai 1984

P.DUHAMEL

Un algorithme de transformation de Fourier rapide à double base.

Annales des télécommunications, pp 481-494, sept-oct 1985

C.ERSKINE & S.MAGAR

Chip's hardware, software speeds up processing in laboratory instrumentation.

Electronic Design, avr. 1985

C.ERSKINE, S.MAGAR, E.CAUDEL, D.ESSIG & A.LEVINSPUHL

Architecture et applications d'un processeur de traitement de signal de la seconde génération : le TMS32020.

Traitement du Signal, vol. 2, n° 1, 1985

R.T.GALLAGHER

A new way to measure submicron line widths.

Electronics, fév. 1986

M.GINDRE & D.HOSTE

Processeur FFT temps réel utilisant la famille Am 29500.

Minis et Micros, n^{OS} 218, 223, 225

M.E.GUILLAUME

Mesure des dimensions critiques dans l'industrie du
semiconducteur.

Echo des recherches, 1986

M.E.GUILLAUME, N.NOAILLY, M.PICHOT, J.L.BUEVOZ

Evaluation of a Fourier transform method for accurate
critical dimension measurement.

Microcircuit Engineering, pp 211-218, 1985

M.E.GUILLAUME, N.NOAILLY, J.C.REYNAUD, J.L.BUEVOZ

Fourier transform method for optical linewidth measurement.

SPIE Proc, Vol 480, pp 71-77, mai 1984

D.JOUVET

Calcul d'une transformée de Fourier rapide sur le MPTS.

Document interne C.N.E.T., fév. 1982

J.LIFERMANN

Les méthodes rapides de transformation du signal.

MASSON, 1979

H.LILEN

Interfaces pour microprocesseurs et micro-ordinateurs.

Editions Radio, 1981

P.LORRAIN

Panorama des circuits et coprocesseurs pour calcul arithmétique.

Minis et Micros, n° 216, pp 107-111

J.MAX

Méthodes et techniques de traitement du signal et applications aux mesures physiques.

MASSON, 1980

N.NOAILLY

MALT, nouvelle méthode de mesure automatique de largeur de traits pour la microélectronique.

Thèse, Grenoble INPG, juillet 1985

Programs for Digital Signal Processing.

IEE Press, New York, 1979

L.R.RABINER, B.GOLD

Theory and Application of Digital Signal Processing.

Prentice-Hall, INC., 1975

J.E.RETSKY

MC68020 & MC68881 Platform board for evaluation in a 16 bit system.

Motorola, Application note, 1985

R.REYNAUD,F.DEVOD & E.MARTIN

Apport de l'intégration pour les processeurs de traitement de signal.

MESURES, nov. 1985

P.H.SINGER

Linewidth measurement : approaching the submicron dimension.

Semiconductor International, pp 48-54, mars 1983

TMS32010 Application brief on FFT.

Texas Instruments, oct 1982

TMS32020 User's Guide.

Texas Instruments, 1985

Digital Signal Processing

Vol.3, Technology and Applications.

Mackintosh Internatonal, 1985

J.VOELKER

New architecture increase speed, and RISC processors are moving from university laboratories into commercial products.

IEE SPECTRUM, janv. 1986

J.L.ZOCESIO

Transformations de Fourier discrètes : algorithmes rapides.

Annales des télécommunications, pp 495-507, sept-oct 1985

P.ABAUZIT

Etude et réalisation d'un calculateur rapide.

Thèse, Grenoble INPG, 1980

ANNEXES

ANNEXE 1

Programmes Fortran de la transformée de Fourier
utilisés dans MALT.

1- Sur HP1000.

2- Sur TMS32020.

```

C
C -----
C SUBROUTINE: FFA
C FAST FOURIER ANALYSIS SUBROUTINE
C -----
C
C SUBROUTINE FFA(B, NFFT)
C
C THIS SUBROUTINE REPLACES THE REAL VECTOR B(K), (K=1,2,...,N),
C WITH ITS FINITE DISCRETE FOURIER TRANSFORM. THE DC TERM IS
C RETURNED IN LOCATION B(1) WITH B(2) SET TO 0. THEREAFTER, THE
C JTH HARMONIC IS RETURNED AS A COMPLEX NUMBER STORED AS
C B(2*J+1) + I B(2*J+2). NOTE THAT THE N/2 HARMONIC IS RETURNED
C IN B(N+1) WITH B(N+2) SET TO 0. HENCE, B MUST BE DIMENSIONED
C TO SIZE N+2.
C SUBROUTINE IS CALLED AS FFA (B,N) WHERE N=2**M AND B IS AN
C N TERM REAL ARRAY. A REAL-VALUED, RADIX 8 ALGORITHM IS USED
C WITH IN-PLACE REORDERING AND THE TRIG FUNCTIONS ARE COMPUTED AS
C NEEDED.
C
C DIMENSION B(2)
C COMMON /CON/ PII, P7, P7TWO, C22, S22, PI2
C
C IW IS A MACHINE DEPENDENT WRITE DEVICE NUMBER
C
C IW = I1MACH(2)
C
C PII = 4.*ATAN(1.)
C PI8 = PII/8.
C P7 = 1./SQRT(2.)
C P7TWO = 2.*P7
C C22 = COS(PI8)
C S22 = SIN(PI8)
C PI2 = 2.*PII
C N = 1
C DO 10 I=1,15
C M = I
C N = N*2
C IF (N.EQ.NFFT) GO TO 20
C
C 10 CONTINUE
C WRITE (IW,9999)
C 9999 FORMAT (30H NFFT NOT A POWER OF 2 FOR FFA)
C STOP
C 20 CONTINUE
C N8POW = M/3
C
C DO A RADIX 2 OR RADIX 4 ITERATION FIRST IF ONE IS REQUIRED
C
C IF (M-N8POW*3-1) 50, 40, 30
C 30 NN = 4
C INT = N/NN
C CALL R4TR(INT, B(1), B(INT+1), B(2*INT+1), B(3*INT+1))
C GO TO 60
C 40 NN = 2
C INT = N/NN
C CALL R2TR(INT, B(1), B(INT+1))
C GO TO 60
C 50 NN = 1
C
C PERFORM RADIX 8 ITERATIONS
C
C 60 IF (N8POW) 90, 90, 70
C 70 DO 80 IT=1,N8POW
C NN = NN*8
C INT = N/NN
C CALL R8TR(INT, NN, B(1), B(INT+1), B(2*INT+1), B(3*INT+1),
C B(4*INT+1), B(5*INT+1), B(6*INT+1), B(7*INT+1), B(1),
C B(INT+1), B(2*INT+1), B(3*INT+1), B(4*INT+1), B(5*INT+1),
C B(6*INT+1), B(7*INT+1))
C 80 CONTINUE
C
C PERFORM IN-PLACE REORDERING
C
C 90 CALL ORD1(M, B)
C CALL ORD2(M, B)
C T = B(2)
C B(2) = 0.
C B(NFFT+1) = T
C B(NFFT+2) = 0.
C DO 100 I=4,NFFT,2
C B(I) = -B(I)
C 100 CONTINUE
C RETURN
C END

```

```

C
C-----
C SUBROUTINE: R2TR
C RADIX 2 ITERATION SUBROUTINE
C-----
C
C
SUBROUTINE R2TR(INT, B0, B1)
DIMENSION B0(2), B1(2)
DO 10 K=1,INT
  T = B0(K) + B1(K)
  B1(K) = B0(K) - B1(K)
  B0(K) = T
10 CONTINUE
RETURN
END

C
C-----
C SUBROUTINE: R4TR
C RADIX 4 ITERATION SUBROUTINE
C-----
C
SUBROUTINE R4TR(INT, B0, B1, B2, B3)
DIMENSION B0(2), B1(2), B2(2), B3(2)
DO 10 K=1,INT
  R0 = B0(K) + B2(K)
  R1 = B1(K) + B3(K)
  B2(K) = B0(K) - B2(K)
  B3(K) = B1(K) - B3(K)
  B0(K) = R0 + R1
  B1(K) = R0 - R1
10 CONTINUE
RETURN
END

C
C-----
C SUBROUTINE: R8TR
C RADIX 8 ITERATION SUBROUTINE
C-----
C
SUBROUTINE R8TR(INT, NN, BR0, BR1, BR2, BR3, BR4, BR5, BR6, BR7,
  BI0, BI1, BI2, BI3, BI4, BI5, BI6, BI7)
DIMENSION L(15), BR0(2), BR1(2), BR2(2), BR3(2), BR4(2), BR5(2),
  BR6(2), BR7(2), BI0(2), BI1(2), BI2(2), BI3(2), BI4(2),
  BI5(2), BI6(2), BI7(2)
COMMON /CON/ PII, P7, P7TWO, C22, S22, PI2
EQUIVALENCE (L15,L(1)), (L14,L(2)), (L13,L(3)), (L12,L(4)),
  (L11,L(5)), (L10,L(6)), (L9,L(7)), (L8,L(8)), (L7,L(9)),
  (L6,L(10)), (L5,L(11)), (L4,L(12)), (L3,L(13)), (L2,L(14)),
  (L1,L(15))

C
C SET UP COUNTERS SUCH THAT JTHET STEPS THROUGH THE ARGUMENTS
C OF W, JR STEPS THROUGH STARTING LOCATIONS FOR THE REAL PART OF THE
C INTERMEDIATE RESULTS AND JI STEPS THROUGH STARTING LOCATIONS
C OF THE IMAGINARY PART OF THE INTERMEDIATE RESULTS.
C
L(1) = NN/8
DO 40 K=2,15
  IF (L(K-1)-2) 10, 20, 30
10  L(K-1) = 2
20  L(K) = 2
  GO TO 40
30  L(K) = L(K-1)/2
40  CONTINUE
PIOVN = PII/FLOAT(NN)
JI = 3
JL = 2
JR = 2
DO 120 J1=2,L1,2
DO 120 J2=J1,L2,L1

DO 120 J3=J2,L3,L2
DO 120 J4=J3,L4,L3
DO 120 J5=J4,L5,L4
DO 120 J6=J5,L6,L5
DO 120 J7=J6,L7,L6
DO 120 J8=J7,L8,L7
DO 120 J9=J8,L9,L8
DO 120 J10=J9,L10,L9
DO 120 J11=J10,L11,L10
DO 120 J12=J11,L12,L11
DO 120 J13=J12,L13,L12
DO 120 J14=J13,L14,L13
DO 120 JTHET=J14,L15,L14
TH2 = JTHET - 2
IF (TH2) 50, 50, 90
50  DO 60 K=1,INT
    T0 = BR0(K) + BR4(K)
    T1 = BR1(K) + BR5(K)
    T2 = BR2(K) + BR6(K)
    T3 = BR3(K) + BR7(K)
    T4 = BR0(K) - BR4(K)
    T5 = BR1(K) - BR5(K)
    T6 = BR2(K) - BR6(K)
    T7 = BR3(K) - BR7(K)

```



```

BR2(K) = T0 - T2
BR3(K) = T1 - T3
T0 = T0 + T2
T1 = T1 + T3
BR0(K) = T0 + T1
BR1(K) = T0 - T1
PR = P7*(T5-T7)
PI = P7*(T5+T7)
BR4(K) = T4 + PR
BR7(K) = T6 + PI
BR6(K) = T4 - PR
BR5(K) = PI - T6
60 CONTINUE
IF (NN-8) 120, 120, 70
70 K0 = INT*8 + 1
KL = K0 + INT - 1
DO 80 K=K0,KL
PR = P7*(BI2(K)-BI6(K))
PI = P7*(BI2(K)+BI6(K))
TR0 = BI0(K) + PR
TI0 = BI4(K) + PI
TR2 = BI0(K) - PR
TI2 = BI4(K) - PI
PR = P7*(BI3(K)-BI7(K))
PI = P7*(BI3(K)+BI7(K))
TR1 = BI1(K) + PR
TI1 = BI5(K) + PI
TR3 = BI1(K) - PR
TI3 = BI5(K) - PI
PR = TR1*C22 - TI1*S22
PI = TI1*C22 + TR1*S22
BI0(K) = TR0 + PR
BI6(K) = TR0 - PR
BI7(K) = TI0 + PI
BI1(K) = PI - TI0
PR = -TR3*S22 - TI3*C22
PI = TR3*C22 - TI3*S22
BI2(K) = TR2 + PR
BI4(K) = TR2 - PR
BI5(K) = TI2 + PI
BI3(K) = PI - TI2
80 CONTINUE
GO TO 120
90 ARG = TH2*PIOVN
C1 = COS(ARG)
S1 = SIN(ARG)
C2 = C1**2 - S1**2
S2 = C1*S1 + C1*S1
C3 = C1*C2 - S1*S2
S3 = C2*S1 + S2*C1
C4 = C2**2 - S2**2
S4 = C2*S2 + C2*S2
C5 = C2*C3 - S2*S3
S5 = C3*S2 + S3*C2
C6 = C3**2 - S3**2
S6 = C3*S3 + C3*S3
C7 = C3*C4 - S3*S4
S7 = C4*S3 + S4*C3
INT8 = INT*8
J0 = JR*INT8 + 1
K0 = JI*INT8 + 1
JLAST = J0 + INT - 1
DO 100 J=J0,JLAST
K = K0 + J - J0
TR1 = BR1(J)*C1 - BI1(K)*S1
TI1 = BR1(J)*S1 + BI1(K)*C1
TR2 = BR2(J)*C2 - BI2(K)*S2
TI2 = BR2(J)*S2 + BI2(K)*C2
TR3 = BR3(J)*C3 - BI3(K)*S3
TI3 = BR3(J)*S3 + BI3(K)*C3
TR4 = BR4(J)*C4 - BI4(K)*S4
TI4 = BR4(J)*S4 + BI4(K)*C4
TR5 = BR5(J)*C5 - BI5(K)*S5
TI5 = BR5(J)*S5 + BI5(K)*C5
TR6 = BR6(J)*C6 - BI6(K)*S6
TI6 = BR6(J)*S6 + BI6(K)*C6
TR7 = BR7(J)*C7 - BI7(K)*S7
TI7 = BR7(J)*S7 + BI7(K)*C7

T0 = BR0(J) + TR4
T1 = BI0(K) + TI4
TR4 = BR0(J) - TR4
TI4 = BI0(K) - TI4
T2 = TR1 + TR5
T3 = TI1 + TI5
TR5 = TR1 - TR5
TI5 = TI1 - TI5
T4 = TR2 + TR6
T5 = TI2 + TI6
TR6 = TR2 - TR6
TI6 = TI2 - TI6
T6 = TR3 + TR7
T7 = TI3 + TI7
TR7 = TR3 - TR7
TI7 = TI3 - TI7

```

C

C

```
TR0 = T0 + T4
TI0 = T1 + T5
TR2 = T0 - T4
TI2 = T1 - T5
TR1 = T2 + T6
TI1 = T3 + T7
TR3 = T2 - T6
TI3 = T3 - T7
T0 = TR4 - TI6
T1 = TI4 + TR6

T4 = TR4 + TI6
T5 = TI4 - TR6
T2 = TR5 - TI7
T3 = TI5 + TR7
T6 = TR5 + TI7
T7 = TI5 - TR7
BR0(J) = TR0 + TR1
BI7(K) = TI0 + TI1
BI6(K) = TR0 - TR1
BR1(J) = TI1 - TI0
BR2(J) = TR2 - TI3
BI5(K) = TI2 + TR3
BI4(K) = TR2 + TI3
BR3(J) = TR3 - TI2
PR = P7*(T2-T3)
PI = P7*(T2+T3)
BR4(J) = T0 + PR
BI3(K) = T1 + PI
BI2(K) = T0 - PR
BR5(J) = PI - T1
PR = -P7*(T6+T7)
PI = P7*(T6-T7)
BR6(J) = T4 + PR
BI1(K) = T5 + PI
BI0(K) = T4 - PR
BR7(J) = PI - T5
100 CONTINUE
JR = JR + 2
JI = JI - 2
IF (JI-JL) 110, 110, 120
110 JI = 2*JR - 1
JL = JR
120 CONTINUE
RETURN
END
```

&FFTE T=00004 IS ON CR00011 USING 00018 BLKS R=0000

```
0001 FTN,L
0002 SUBROUTINE FFTE (TRANS,NFFT)
0003 C
0004 C VERSION DE MARS 1986
0005 C *****
0006 C
0007 C *****
0008 C * FFTE PERMET DE CALCULER LA TRANSFORMEE DE FOURIER D'UN SIG
0009 C * REEL DONT LE NOMBRE D'ECHANTILLONS EST UNE PUISSANCE DE 2
0010 C *****
0011 C
0012 C LA SEQUENCE D'APPEL EST:
0013 C CALL FFTE(TRANS,NFFT)
0014 C TRANS = TABLEAU CONTENANT LES VALEURS DE LA
0015 C NFFT = NOMBRE D'ELEMENTS COMPLEXES A CALCUL
0016 C
0017 C EN ENTREE : LES DONNEES SONT RANGEES DANS I
0018 C
0019
0020 COMMON IX(1024)
0021 DIMENSION TRANS(1024)
0022 PI=4.*ATAN(1.)
0023 C *****
0024 C * BIT REVERSED *
0025 C *****
0026 NN=NFFT/2
0027 DO 5 I=1,NN
0028 TRANS(2*I-1)=FLOAT(IX(2*I-1))
0029 TRANS( I )=FLOAT(IX(2*I))
0030 S CONTINUE
0031 J=1
0032 DO 80 I=1,NN
0033 IF(I.LT.J) THEN
0034 XTR=TRANS(2*J-1)
0035 XTI=TRANS(2*J)
0036 TRANS(2*J-1)=TRANS(2*I-1)
0037 TRANS(2*J)=TRANS(2*I)
0038 TRANS(2*I-1)=XTR
0039 TRANS(2*I)=XTI
0040 ENDIF
0041 M=NN/2
0042 50 IF(J-M) 70,70,60
0043 60 J=J-M
0044 M=(M+1)/2
0045 GOTO 50
0046 70 J=J+M
0047 80 CONTINUE
0048 C *****
0049 C * CALCUL DE LA FFT COMPLEXE SUR NFFT/2 POINTS *
0050 C *****
0051
0052 MMAX=1
0053
0054 90 IF (MMAX-NN) 100,130,130
0055 100 ISTEP=2*MMAX
0056 DO 110 M=1,MMAX
0057 COEF=FLOAT(1-M)/FLOAT(MMAX)
0058 THETA=PI*COEF
0059 C=COS(THETA)
```

```

0061
0062          DO 120 J=1,NN, ISTEP
0063             L=I+MMAX
0064             XTR=TRANS(2*L-1)*C-TRANS(2*L)*S
0065             XTI=TRANS(2*L)*C+TRANS(2*L-1)*S
0066             TRANS(2*L-1)=TRANS(2*I-1)-XTR
0067             TRANS(2*L)=TRANS(2*I)-XTI
0068             TRANS(2*I-1)=TRANS(2*I-1)+XTR
0069             TRANS(2*I)=TRANS(2*I)+XTI
0070          120      CONTINUE
0071          110      CONTINUE
0072             MMAX=ISTEP
0073             GOTO 90
0074          130      CONTINUE
0075
0076      C .....
0077      C      GOTO 1000
0078      C .....
0079      C      *****
0080      C      * DERNIER PAPILLON DE LA FFT REELLE *
0081      C      *****
0082             TRANS(1)=TRANS(1)+TRANS(2)
0083             TRANS(2)=0
0084             DO 200 K=2, NN/2
0085                 L=NN-K+2
0086                 COEF=FLOAT(1-K)/FLOAT(NN)
0087                 THETA=PI*COEF
0088                 C=COS(THETA)
0089                 S=SIN(THETA)
0090                 GRK=(TRANS(2*K-1)+TRANS(2*L-1))/2
0091                 GIK=(TRANS(2*K)-TRANS(2*L))/2
0092                 HIK=(TRANS(2*L-1)-TRANS(2*K-1))/2
0093                 HRK=(TRANS(2*L)+TRANS(2*K))/2
0094                 HKWI=C*HIK+S*HRK
0095                 HKWR=C*HRK-S*HIK
0096                 TRANS(2*K-1)=GRK+HKWR
0097                 TRANS(2*K)=GIK+HKWI
0098                 TRANS(2*L-1)=GRK-HKWR
0099                 TRANS(2*L)=HKWI-GIK
0100          200      CONTINUE
0101             TRANS(NN+2)=-TRANS(NN+2)
0102             RETURN
0103             END

```

ANNEXE 2

Comparaison des performances des microprocesseurs
de calcul rapide actuellement disponibles.

Opérateurs multiformats ou multifonctions

Fabricant	Référence	Formats (*)	Opérations	Performances	Consomm. max. (W)	Techno.	Nbre broches	Disponib.	Remarques					
AMD	9511 A	F 32 E 16, E 32	Add., Sous., Mul., Div., Log., Exp., Trigo, Racine carrée	à 4 MHz Add./Sous.. Mul. Div. en µs	+ 12V: 1,2W + 5V: 0,5W	N-Mos	24	Oui	Format F 32 incompatible IEEE. Interface microprocesseur 8 bits					
	9512	F 32, F 64	Add., Sous., Mul., Div.	à 3 MHz Add./Sous.. Mul. Div. en µs						+ 12V: 1,2W + 5V: 0,5W	N-Mos	24	Oui	Interface microprocesseur 8 bits
	29325	F 32	Add., Sous., Mul.	10 Mop/s						6 ?	bipolaire	144	Oct. 84	Trois bus 32 bits Division flottante en 8 cycles
Analog Devices	ADSP 3210	F 32, F 64 E 32	Mul.	F 32 : 10 Mop/s F 64 : 2 Mop/s E 32 : 10 Mop/s	0,4	C-Mos	96	Début 85	Deux bus 32 bits					
	ADSP 3220	E 32	Add., Sous.,		0,4		96							
Intel	8087	F 8, E 16, E 32, E 64, F 32, F 64, F80 BCD	Add., Sous., Mul., Div., Log., Exp., Tang, Racine carrée	à 8 MHz Add./Sous.. Mul. Div. en µs	2,6	H-Mos	40	Oui	Format interne 80 bits. Interface 8/16 bits					
	80287	E 16, E 32, E 64, F 32, F 64, BCD	Add., Sous., Mul., Div., Log., Exp., Tang, Racine carrée	à 5 MHz Add./Sous.. Mul. Div. en µs						2,6	H-Mos	40	Oui	Version définitive disponible fin 1984 Version 8 MHz fin 1984
MMI	74 S 508	E 8	Mul., Div.,	Mul. : 08 µs Div. : 2,2 µs	2	bipolaire	24	Oui	Multiplieurs/diviseurs séquentiels					
	74 S 516	E 16	Mul., Div.	Mul. : 1,5 µs Div. : 3,5 µs	2,5	bipolaire	24	Oui						
Motorola	68881	F 32, F 64, F80, BCD	Add., Sous., Mul., Div., Trigo, Log., Exp., Racine carrée	<i>mul. 74 µs div 61 µs bin 24 µs</i>	?	N-Mos ?	64	Début 85	Format interne 80 bits. Nombreuses instructions. Interface 8, 16, 32 bits					
NS	16081	F 32, F 64	Add., Sous., Mul., Div.	à 10 MHz Add./Sous.. Mul. Div. en µs	1,5	N-Mos	24	Oui (version 6 MHz)	Interface 16 bits. Version 10 MHz : juillet 1984					
TI	TMS 32010	E 8, E 16, E 32	Add., Sous., Mul.	Tcycle = 200 ns Mul. 16 x 16 = 200 ns	1,5	N-Mos	40	Oui	Rom et Ram interne. Décaleur à barillet ALU 32 bits. Multiplieur 16 x 16. Interruption					
Weitek (représ. par Franelec)	WTL 1033	F 32	Add., Sous.,	8 Mop/s	2,2	N-Mos	64	Oui	Version 10 Mop/s : juillet 1984					
	WTL 1032		Mul.		2,5		64	Oui						
	WTL 1065	E 32	Add., Sous.,	Add./Sous. : 8 Mop/s Mul. F 32 : 4 Mop/s Mul. F 64 : 2 Mop/s	3,3	N-Mos	144	Fin 84	Fonctionnement possible sur 1, 2, ou 3 bus					
	WTL 1064	F 32, F 64	Mul.	3,3	144		Fin 84							
Zilog	8070	E 16, E 32, F16, 64, 80, BCD	Add., Sous., Mul., Div., Racine carrée	à 10 MHz ADD. Mul. Div. en µs	?	N-Mos	64	Fin 85						

(*) Formats : E et F suivis d'un nombre indiquent des entiers ou flottants et le nombre de bits. BCD = binaire codé décimal.

SOME AVAILABLE DSP BUILDING BLOCK CIRCUITS

SUPPLIER	DEVICE	FUNCTION	CYCLE TIME	TECHNOLOGY
AMD	29-L/C/510	16-bit Multiplier Accumulator	70/100/140ns	Bipolar/CMOS
	29325	32-bit FP ALU & Multiplier	100ns	Bipolar
	29501	Micro Programmable Signal Processor	400ns	Bipolar
	29509	12-bit Multiplier/Accumulator		Bipolar
	29516A/17A	16-bit Multiplier	45ns	Bipolar
	29520/21	Multi-level Pipeline Register		Bipolar
	29540	16-bit FFT Address Sequencer		Bipolar
	29C516/7	16-bit Multiplier	125ns	CMOS
	29L516/7	16-bit Multiplier	85ns	Bipolar
ANALOG DEVICES	1101	16-bit Multiplier Accumulator	60ns	CMOS
	1201	16(32) bit ALU and Shift Unit	60ns	CMOS
	1401	Program Sequencer	30ns	CMOS
	1410	16-bit Address Generator	30ns	CMOS
	3210	32-bit FP Multiplier	100ns	CMOS
	3220	32-bit FP ALU	100ns	CMOS
	ADSP 1008A	8-bit Multiplier/Accumulator	45ns	CMOS
	ADSP 1009	12-bit Multiplier/Accumulator	150ns	CMOS
	ADSP 1010A	16-bit Multiplier/Accumulator	190ns	CMOS
	ADSP 1012	12-bit Multiplier	130ns	CMOS
	ADSP 1016	16-bit Multiplier	170ns	CMOS
	ADSP 1024	24-bit Multiplier	235, 275ns	CMOS
	ADSP 1200	ALU Shifter	75ns	CMOS
IHI	MPY-121HI	12-bit Multiplier (gate array)	150ns	CMOS
MATSUSHITA	MC 9981	Floating Point Multiplier	80ns	CMOS
NEC		16-bit Multiplier	45ns	CMOS
ROCKWELL		16-bit Multiplier	120ns	CMOS/SOS
SYNERTEK	SY66016B/A/-	16-bit Multiplier	90ns/150/200	NMOS
TEXAS INSTRUMENTS	74ALS1616	16-bit Multiplier	55ns	Bipolar
	74AS888/90	8-bit ALU/Microcontroller		Bipolar
	74AS897	Address Generator		Bipolar
	TMS230	16-bit Microprocessor/Multiplier		NMOS
TOSHIBA	T6386/7	Microprocessor/Multiplier		CMOS
	T7429	16-bit Multiplier	27ns	CMOS/SOS
TRW	MPY 008H	8-bit Multiplier	90ns	CMOS
	MPY 012H	12-bit Multiplier	115ns	CMOS
	MPY 016H	16-bit Multiplier	145ns	Bipolar
	MPY 016K	16-bit Multiplier	45ns	Bipolar
	TDC 1006	256 x 1 Shift Register	40ns	
	TDC 1011	18 x 8 Programmable Delay Register	55ns	
	TDC 1022	22-bit Arithmetic Unit		Bipolar
	TDC 1033	RALU		
	TDC 1042	Floating Point Multiplier		
	TMC 1030	64 x 9 bit FIFO	66ns	Bipolar
TMC 2008	8-bit Multiplier/Accumulator	100ns	CMOS	
TMC 2009	12-bit Multiplier/Accumulator	135ns	CMOS	
TMC 2010	16-bit Multiplier/Accumulator	165ns	CMOS	
WEITEK	WTL 1010A/-/L	16-bit Multiplier/Accumulator	90/120/140ns	NMOS
	WTL 1016/A	16-bit Multiplier	140/100ns	NMOS
	WTL 1032/33	32-bit FP Multiplier & Adder	200ns	NMOS
	WTL 1064/5	Multiplier/ALU	60ns	NMOS
	WTL 1516/A	Microprocessor/Multiplier	140/90ns	NMOS

SINGLE CHIP PROGRAMMABLE DIGITAL SIGNAL PROCESSORS

DEVICE	AMI 52811	AMI 28211/2	DELL DSP-1	FUJITSU MB 8764	HITACHI 61810	INTEL 2920/21	NEC 7720/P20	NEC 77220	TI 320	TOSHIBA 16386/7
Process	NMOS	NMOS	NMOS	CMOS	CMOS	EPRM/NMOS	NMOS/EPRM	CMOS	NMOS	CMOS
Minimum Feature Size	4.5 microns	3 microns	4.5 microns	2.3 microns	3 microns	6/4 microns	3 microns	2 microns	2.7 microns	2 microns
Year Described	1978	1983	1980	1983	1982	1978/81	1980	1984	1982	1983
Area (sq.mils)	41,000	-	106,000	145,000	79,000	47,000	44,000	-	70,000	76,000/74,000
Number of Transistors	30,000	-	45,000	91,000	55,000	20,000	40,000	-	-	66,030/48,000
Pins	28	28/64	40	88	40	28	28	-	40	28/64
Power	1W	0.7W	1.25W	0.290W	0.200W	1W	1W	-	0.9W	0.360W
Precision	16	16	20	16	16 FP	25	16	24	16	16
Multiplier	12 x 12 x 16	12 x 12 x 16	$\frac{4(4 \times 20)}{36}$	16 x 16 x 26	12 x 12 x 16	25 x 1 = 25	16 x 16 x 31	24 x 24 x 48	16 x 16 x 31	16 x 16 x 31
Speed	300 ns	300 ns	4(200) ns	100 ns	250 ns	600/400 ns	250 ns	100 ns	200 ns	250 ns
Program Memory	256 x 17 ROM	512 x 18 ROM/EXT	1K x 16 ROM, EXT	1K x 24 ROM, EXT	512 x 22 ROM	192 x 24 LPRM/ROM	512 x 23 ROM/EPRM	4096 ROM	4K x 16 ROM, EXT	512 x 16 ROM/EXT
Data RAM	128 x 16	256 x 16	128 x 20	256 x 16	200 x 16	40 x 25	128 x 16	1024 x 24	144 x 16	128 x 16
Data ROM	128 x 16	128 x 16	In program	In program	128 x 16	16 x 4	512 x 13	1924	In program	512 x 16 ROM/EXT

COMPETITIVE FEATURES OF DSP CHIPS

FEATURE	TI TMS 32010	NEC μ PD7720	AMI 52811	Intel 2920
Data-word size (bits)	16	16	16	25
Coefficient size (bits)	16	13	16	Variable
Accumulator width (bits)	32	16	16	28
Saturation arithmetic	Hardware	Software	Hardware	Hardware
Boolean logic operations	Yes	Yes	No	Yes
Multiplier implementation	Hardware	Hardware	Hardware	Software
Multiplier precision (in x in = out)	16 x 16 = 32	16 x 16 = 32	12 x 12 = 16	12 x 25 = 28
Multiplication time (ns) (worst case)	200	250	300	4800
Parallel I/O (bits)	16	8	8	4 in/8 out
Instruction word (bits)	16	23	17	24
Instruction cycle (ns)	200	250	300	400
Subroutine levels	50	4	1	None
Iteration (loop) counter	Yes	No	Yes	No
Conditional jumps	Yes	Yes	Yes	No
Full speed external memory expansion	Yes	No	No	No
Program RAM	Yes	No	No	No
Instruction ROM (bits)	1536 x 16	512 x 23	256 x 17	192 x 24
Coefficient ROM (bits)	Not required	512 x 13	120 x 16	-
Data RAM (bits)	144 x 16	128 x 16	128 x 16	40 x 25
Z ⁻¹ function	Yes	Yes	Yes	No
Lookup tables	Yes	Yes	Yes	No
Package	40-Pin DIP	28-Pin DIP	28-Pin DIP	28-Pin DIP
Interrupts - Hardware	Yes	-	-	-
- Software	Yes	-	-	-

Source: Computer Design

BENCHMARK COMPARISON TMS32010 vs. μ PD7720

	TMS 32010	μ PD ⁺ 7720	Unit
Biquad Filter	1.8	2.25	μ S
Sine or Cosine	4.8	5.25	μ S
μ /A Law to Linear Conversion	0.8	0.5	μ S
32-Point Complex FFT	0.254	0.7	mS
64-Point Complex FFT	0.555	1.6	mS
Normalized Total	1.0	2.7	

DSP GENERATIONS

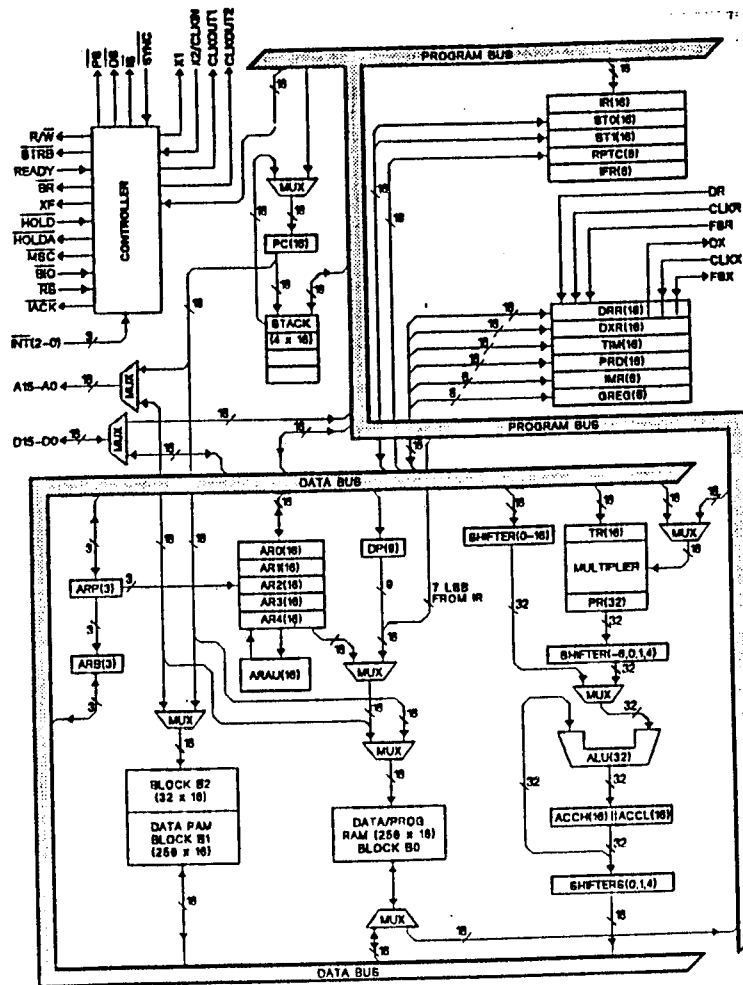
* Benchmarks from NEC Electronics' Digital Signal Processor Data Sheet

GENERATION	MIPS	CATEGORY	CHARACTERISTICS	TYPICAL DEVICES
0	Sub 2 MIPS			Intel 2920
1	5 MIPS or Less	A	16 bit word fixed point	TMS 320-10
		B	Floating point	Hitachi
2	5-10 MIPS	A	Fixed point 16 and 32 bits	Fujitsu
		B	Floating point	TMS 320-20
3	10-20 MIPS	A	Fixed point	
		B	Floating point	

ANNEXE 3

Bloc diagramme du TMS32020.

Caractéristiques du TMS32020.



NOTE:

ACCH = Accumulator high	DRR = Serial port data receive register	PR = P register
ACCL = Accumulator low	DXR = Serial port data transmit register	PRD = Period register for timer
ARAU = Auxiliary register arithmetic unit	GREG = Global memory allocation register	TR = T register
ARB = Auxiliary register pointer buffer	IFR = Interrupt flag register	TIM = Timer
ARP = Auxiliary register pointer	IMR = Interrupt mask register	ST0, ST1 = Status registers
DP = Data memory page pointer	IR = Instruction register	RPTC = Repeat instruction counter

Bloc diagramme du TMS 32020

Spécifications du TMS 32020

- 64 K mots de mémoire de "données", 64 K mots de mémoire de programme.
- Mémoire RAM de "données" /programme de 544x16 bits.
- Temps de cycle 200ns.
- Opération de multiplication/addition en un seul cycle.
- Unité arithmétique et logique (ALU) de 32 bits avec plusieurs options de décalage.
- Cinq registres d'adresse associés à une arithmétique spécialisée.
- Registres de comptage câblés.
- Interface multiprocesseur avec possibilité de synchronisation de l'horloge.
- "Port" série 8/16 bits.
- Interface parallèle 16 bits pour "données", programme et

accès aux entrées/sorties avec l'option "état d'attente".

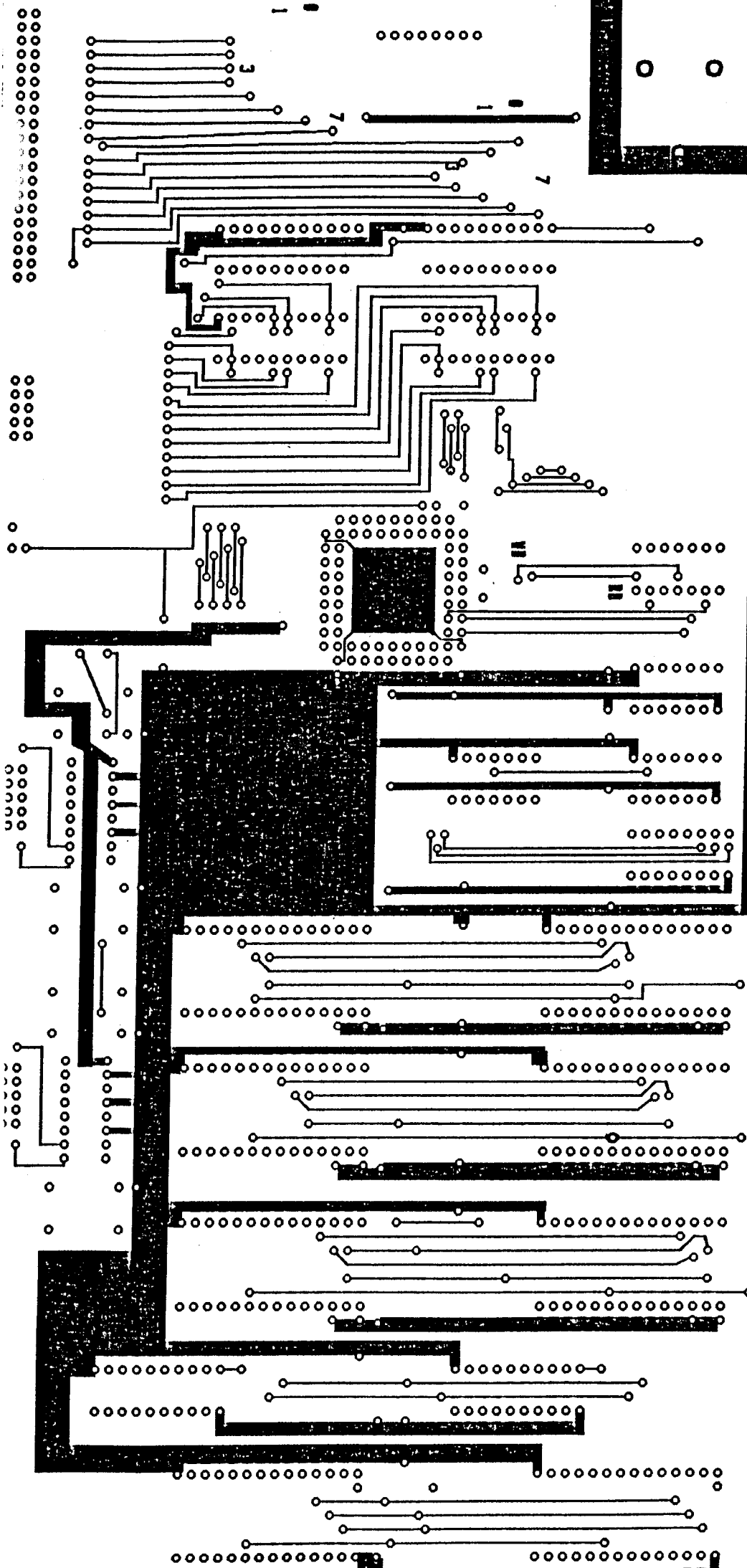
- Opérations arithmétiques en virgule flottante.

- Boîtier 68 broches PGA (Pin Grid Array).

- Alimentation 5 V.

ANNEXE 4

Dessin de la carte JAG sur machine de DAO CALMA.



○ ○
○ ○
○ ○
○ ○

○ ○ ○ ○ ○ ○ ○ ○
○ ○ ○ ○ ○ ○ ○ ○

○ ○ ○ ○ ○ ○ ○ ○
○ ○ ○ ○ ○ ○ ○ ○

MARS 86

