الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

14/85

وزارة التعليم و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT: **électronique**

PROJET DE FIN D'ETUDES

SUJET

Etude et Mise au Point d'un Logiciel de Tracé de Surfaces Mathématiques Sur M6800

Proposé par :
A. ABDELLAH KHODJA
F. ARFI

Etudié par :
BOUCHLAGHEM. Y.
HOCINE S.

Dirigé par :
F. ARFI
F. GUETTACHE

PROMOTION : JANVIER 1985

E.N.P. 10, Avenue Hacen Badi – EL-HARRACH – ALGER

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT: electronique

PROJET DE FIN D'ETUDES

SUJET

Etude et Mise au Point d'un Logiciel de Tracé de Surfaces Mathématiques Sur M6800

Proposé par :
M. A. ABDELLAH-KHODJA
M. F. ARFI

Etudié par :
M. Y. BOUCHLAGHEM
Melle. S. HOCINE

Dirigé par :
M. F. ARFI
M. F. GUETTACHE

PROMOTION : JANVIER 1985

E.N.P. 10, Avenue Hacen Badi – EL-HARRACH – ALGER

Dédicaces

A la mémoire de mon grand-père et de Saâdia
A mes parents et mes frères
Aux Lahmidi
A mes amis

yassine

A la mémoire de ma mère,
A mon père, à ma famille, à mes amis.

Selma.

Remerciements

C'est cordialement que nous remercions :

- MM. TATAH et Bettayeb pour nous avoir accueillis dans leur centre : CDCE.

- Mr ABDELLAH KHODJA pour le sujet qu'il a proposé, pour la confiance placée en nous, pour l'aide nécessaire qu'il nous a apportée ainsi que pour les conseils qu'il n'a cessé de nous prodiguer

- MM. ARFI et GUETTACHE pour leur disponibilité, leur suivi, Leurs soutien moral et conseils avisés.

- MM. ZOUAOUI et SELLAL pour leur aide amicale.
- Mr OUSSAMA pour l'interêt qu'il a porté à ce projet.
- La direction du CNI pour nous avoir permis l'accés à un laboratoire et ainsi d'élaborer notre travail.

Que tous ceux qui ont contribué, de prés ou de loin, à notre formation trouvent ici l'expression de notre reconnaissance et profonde gratitude.

Sommaire

C'est par la force de l'image que par la suite des temps pourraient bien s'accomplir les vraies révolutions.

ANDRE BRETON.

Le poids des mots, le choc des photos: une image vaut mille mots.

AUTEURS.

Introduction

La valeur de l'image comme moyen de communiquer de l'information rapidement et avec precision est reconnue depuis longtemps.

Tout au long de l'histoire, diverses méthodes d'impression, de reproduction et de photographie ont été développées pour créer des motifs visuels. Aujourd'hui, notre civilisation Croule sous la masse des images générées et traitées par ordinateur.

Les applications de l'informatique graphique couvrent un domaine de plus en plus large qui s'étend du simple jeu à la C.A.O la plus élaborée.

Nous citerons notamment :

* Les graphiques financiers et commerciaux.

Ils permettent de présenter rapidement, les relations liant des informations plus ou moins diverses : ce sont des courbes, des diagrammes à barre ou sectoriels ....

La conception assistée par ordinateur

* Que ce soit en architecture, en génie civil ou électrique, en construction navale, automobile ou aéronautique, les images

figuratives constituent le moyen d'expression privilégié des techniques de C.A.O

Il est possible de visualiser un objet sous n'importe quel angle en spécifiant à l'ordinateur les dimensions de la partie à visualiser.

L'objet en cours de tracé peut être observé et des vérifications immédiates sur ses caractéristiques sont entreprises au vu de l'image présentée.

* Le graphique scientifique ou technique.

Sa fonction première est de visualiser un résultat de calculs ou de relevés de mesures.

C'est une aide au diagnostic ou à l'analyse.

En effet, dans beaucoup de disciplines, des spécialistes manipulent des équations de fonctions de deux variables, sans pouvoir imaginer l'allure des surfaces correspondantes. Cependant les spécialistes peuvent tirer des indications importantes sur certaines propriétés, par exemple, à la vue de 'puits' en ce qui concerne le champ créé par une charge, ou autres interprétations physiques.

BUT DU PROJET

L'objectif de ce projet est d'étudier et de mettre au point un logiciel interactif de tracé de surfaces mathématiques sur l'EXORCISER MOTOROLA. Ce logiciel devra traiter aussi bien les surfaces définies explicitement par des fonctions de deux variables du type $y = f(x, z)$ que celles générées par des tableaux de valeurs expérimentales.
Le tracé sera effectué avec elimination des parties cachées sur la table traçante numérique TEKTRONIX 4662
L'etude que nous allons présenter comportera cinq chapitres.

Le premier chapitre introduira les éléments essentiels du problème de la visualisation tridimentionnelle de surfaces d'une part, et présentera les algorithmes de traitement des parties cachées les plus utilisés.

Le second chapitre sera consacré à une présentation détaillée de l'algorithme que nous proposons.

Dans le troisième chapitre, l'algorithme sera mis en œuvre en BASIC sur le système graphique TEKTRONIX 4051.

Quant au quatrième chapitre, il comportera deux parties :

- La première sera consacrée à l'étude et réalisation de la carte d'interface EXORCISER - table traçante ;
- la seconde donnera les détails de l'ecriture du logiciel en assembleur

Enfin, le dernier chapitre sera consacré aux résultats et aux commentaires sur les performances de l'algorithme.

CHAPITRE 1
VISUALISATION D'UNE FONCTION DE DEUX VARIABLES

1 - Position du problème

Pour visualiser une fonction de deux variables sur un écran, nous devons projeter l'éspace à trois dimensions sur le plan de telle manière que la perspective soit représentée. L'effet de perspective dépend alors, de l'orientation du système d'axes choisie.

Un observateur, placé en un certain point de l'éspace, ne peut voir qu'une partie de la surface (supposée opaque) représentant la fonction.

En éliminant les parties ,cachées, on recrée l'illusion du relief, augmentant ainsi considérablement le réalisme de l'image qui serait, sinon, atténué à cause de l'effet de transparence.

Aussi, le traitement à mettre en oeuvre se décompose-t-il en deux parties :

- Passage de l'espace à trois dimensions vers l'espace à deux dimensions de façon à recréer l'effet de perspective.
- Elimination des parties ,cachées.

Plusieurs méthodes ont vu le jour permettant une visualisation dans l'espace à trois dimensions de surfaces représentant des fonctions bivariables.

Dans la première méthode proposée par Kubert et Al (1968, [1]) la visibilité d'un point était déterminée en traçant le segment reliant ce point à l'observateur.

Un point était alors Considéré comme :

- non visible si ce point coupe la surface en plusieurs points.
- visible s'il coupe la surface en un point unique.

Cet algorithme, malgré son extrême rigueur mathématique, présente l'inconvenient de necessiter des calculs longs et difficiles à implémenter sur microprocesseurs.

Les deux algorithmes les plus utilisés à l'heure actuelle sont celui de Williamson [2] et celui de Wright [3] où le test de visibilité se fait directement dans le plan UV et non plus dans l'espace.

2 - L'Algorithme de Williamson

A partir de l'infinité de points constituant la surface représentant la fonction $y = f(x, z)$, on constitue un sous-ensemble permettant de la visualiser. On compose ensuite un tableau de valeurs en Coupant la surface par des plans parallèles au plan xOy, c'est à dire des plans $z = Cte$.

Les courbes ainsi obtenues sont reportées dans un plan après leur avoir fait subir un décalage afin de simuler la profondeur. Cet artifice permet de donner une bonne impression de profondeur, alors qu'on n'utilise qu'une simple perspective cavalière. Le principe est le suivant:

- On Commence par afficher la courbe qui est la plus proche de l'observateur. Cette Courbe sert à initialiser ce que nous nommerons 'fonction visuelle maximale' ou encore 'ligne de crêtes; qui est l'ensemble des points ayant l'ordonnée la plus élevée.

- On prend ensuite la Courbe se trouvant immédiatement derrière la première, par rapport à l'observateur et on Compare chaque point de la nouvelle Courbe aux valeurs de la fonction visuelle maximale. Chaque fois que pour un x donné, on trouve un y inferieur au y correspondant de la ligne de crêtes, ceci signifie que le point est caché.
Chaque fois que l'on trouve un point tel que le y soit superieur au y correspondant de la ligne de crêtes, cela signifie qu'il faut afficher le point en question et mettre à jour la fonction visuelle maximale en ce point.

- On Continue le processus de comparaison puis d'affichage et de mise à jour jusqu'à ce que tous les plans de coupe aient

été explorés.

Pour réaliser cet algorithme, on utilise un tableau contenant toutes les coordonnées des points formant la ligne de crêtes. Pour chaque point de la courbe à afficher, on regarde s'il y a un x correspondant dans le tableau "ligne de crêtes". (Celui-ci est trié en ordre croissant sur les x pour accelérer la recherche.) Si on ne trouve pas de x correspondant au point désiré, on trouvera cependant deux points x_i et x_{i+1} de la ligne de crêtes tels que $x_i \leq x \leq x_{i+1}$. On calcule alors par interpolation linéaire entre y_i et y_{i+1} la valeur de la fonction visuelle maximale au point x. Si cette valeur est superieure au y d'entrée, le point est caché, sinon il y a mise à jour de la liste des points formant la ligne de crêtes par insertion du point (x,y). Notons au passage, que généralement on maintient à jour une fonction visuelle minimale qui permet de représenter aussi le dessous de la surface. Le principe est exactement le même que pour la fonction visuelle maximale. Nous pouvons faire les remarques suivantes:

- Cet algorithme pousse les calculs jusqu'à la précision de la machine. Ceci signifie que dans la liste "ligne de crêtes" peuvent figurer de nombreux points qui seront confondus lors de l'affichage, du fait de la résolution limitée de l'écran;

- Williamson demande à l'utilisateur d'évaluer la taille des tableaux qui contiendront la fonction visuelle maximale et la fonction visuelle minimale, ce qui est assez difficile.

En résumé, la méthode de Williamson, qui était un progrès très grand par rapport à la première méthode proposée, est cependant mal adaptée à l'utilisation d'un terminal graphique puisqu'elle ne tient pas compte de ses caractéristiques.

3 - L'Algorithme de Wright

Si le point de départ est commun : découpage de la surface par des plans à $z = Cte$, utilisation de lignes de crêtes maximale et minimale, deux différences essentielles apparaissent :

- Utilisation d'une projection perspective pour la représentation de l'espace, permettant en particulier un traitement simple des courbes à $x = Cte$ sur la même figure ;

- Utilisation de la faiblesse de résolution de l'écran, pour ne considérer que ce qui se passe aux points de coordonnées entières. La fonction visuelle est alors contenue dans un tableau de taille fixe, contenant autant d'éléments qu'il y a de points sur une ligne d'écran (512 ou 1024 pour les cas les plus courants). Les deux tableaux "Ligne de crêtes" contiennent dans chaque élément, l'ordonnée du point le plus haut de tous les points de même abscisse.

Le problème de la recherche d'un élément est donc évité (on se contente d'une indexation) et il n'y a jamais d'insertion dans la liste. Les seules opérations consistent éventuellement à mettre à jour la fonction visuelle entre deux points x_i et x_j non adjacents en calculant par interpolation linéaire les valeurs aux points $x_{i+1}, \ldots, x_{j-1}$. L'algorithme est alors suffisamment performant pour que l'on envisage de présenter à la fois les courbes à z = cte et à x = Cte. Le seul problème est que l'on ne peut pas se contenter de superposer le tracé des courbes à z = cte et les courbes à x = Cte.

Il faut alors opérer de la façon suivante :

- On trace la j-ième courbe à z = cte ;
- On trace alors en chaque point le morceau de courbe à x = cte compris entre la j-ième et la j+1-ième courbe à z = cte, en mettant à jour les lignes de crêtes ;
- On trace alors la j+1-ième courbe à z = cte.

L'algorithme proposé, que nous développerons au chapitre 2 est une synthèse des deux. Il se caractérise notamment par :

- l'utilisation d'une perspective cavalière
- des lignes de crêtes de taille fixe et définies comme suit :
  * la ligne de crêtes maximale est l'ensemble de points ayant l'ordonnée la plus élevée.
  * la ligne de crêtes minimale est l'ensemble de points

ayant l'ordonnée la plus faible.

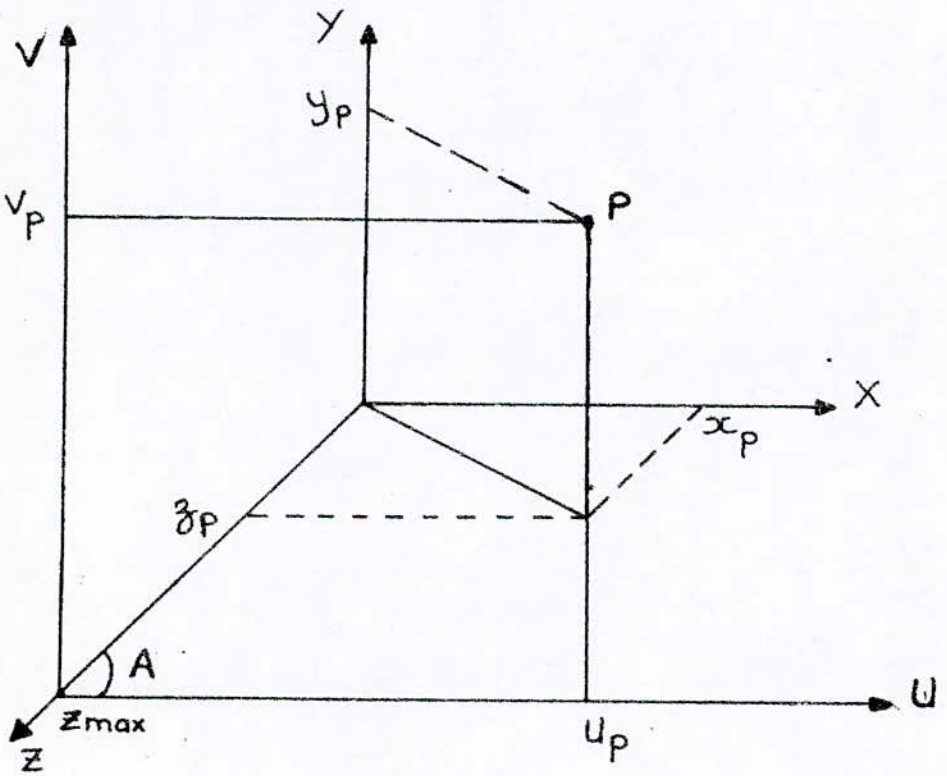
Remarque:
Les lignes de crêtes sont des lignes fictives qui n'appartiennent à aucun plan particulier.

CHAPITRE 2
ALGORITHME

1 Représentation de l'espace à trois dimensions dans le plan de visualisation

Le plan de représentation peut être décrit au moyen de deux axes U et V. Tout point de ce plan peut être représenté par ses coordonnées dans ce système d'axes.

Afin de tracer la fonction $y = f(x,z)$, il suffit de déterminer pour chaque point de la surface décrite par cette équation, ses coordonnées U et V.



- figure 1 -

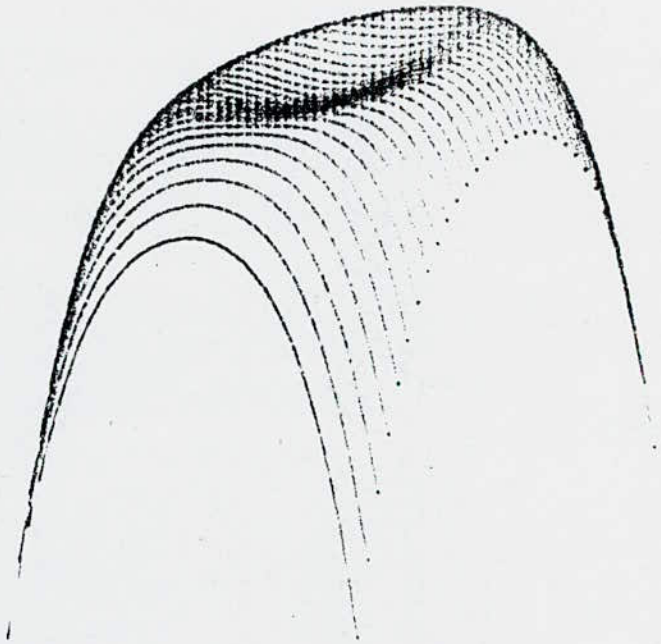
On choisit de placer le repère X, Y, Z de telle façon que le point (X=0, Y=0, Z = Zmax) coïncide avec l'origine des axes U et V. Cela revient à dire que pour la courbe la plus proche de l'observateur, le plan XOY se confond avec le plan UV. Un point P de l'espace ayant pour coordonnées X_p, Y_p et Z_p dans le repère X, Y, Z a - dans le plan UV - les coordonnées U_p et V_p. Celles-ci se déduisent des premières par les relations :

$$U_p = (Z_{max} - Z_p) * COS A + X_p$$
$$V_p = (Z_{max} - Z_p) * SIN A + Y_p \qquad 2.1$$

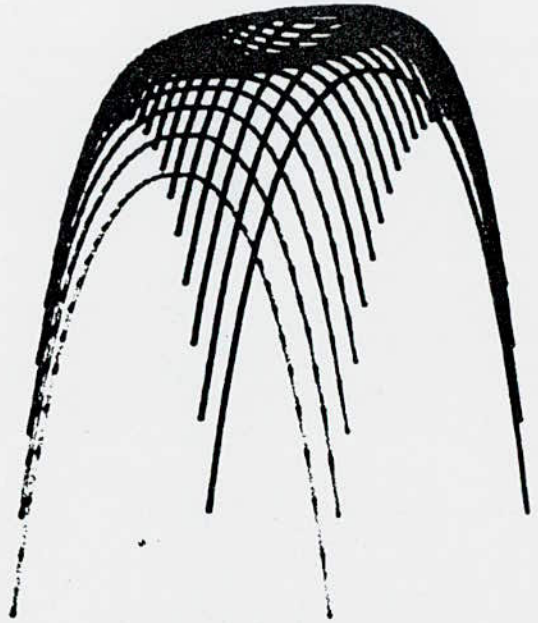
L'effet de perspective sera donné par l'angle que fait l'axe Z avec le plan XOY ou angle de Visée.
L'effacement des parties cachées dans un affichage à trois dimensions indique la profondeur et augmente son réalisme. (fig. 2)

2 - Traitement des parties cachées

La surface étant exprimée par $y = f(x, z)$, elle sera représentée par une série de Courbes.
Pour Z fixé, on trace $y = f_z(x)$ puis on décrémente Z et on trace une deuxième Courbe $y = f_z(x)$ et ainsi de suite jusqu'aux limites.



Avec test des lignes Cachées.



Sans test des lignes Cachées.

A = 45

- figure 2 -

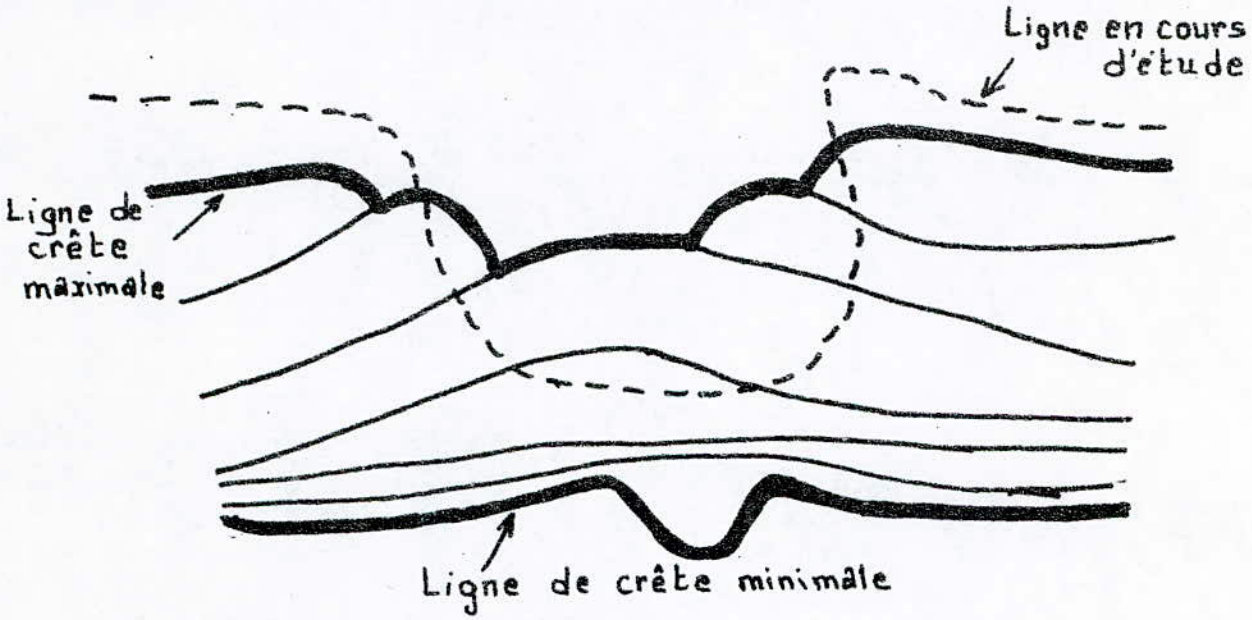
Une fois calculées les coordonnées U et V d'un point, on décide de sa visibilité en comparant son ordonnée à celles des points des lignes de crêtes minimale et maximale ayant même abscisse dans le plan de représentation (figure 3)

Il sera considéré comme :

- Visible
  * si son ordonnée est supérieure à celle du point Correspondant de la ligne de crêtes maximale. Il deviendra alors, un point de cette ligne qui se trouvera ainsi mise à jour. Dans ce cas, il appartiendra à la surface supérieure.
  * si son ordonnée est inférieure à celle du point Correspondant de la ligne de crêtes minimale. Il deviendra alors, un point de cette ligne qui se trouvera ainsi mise à jour. Dans ce cas, il appartiendra à la surface inférieure.
- Non visible
  * si son ordonnée est comprise entre celles des points correspondants des deux lignes de crêtes.

Remarque :

l'utilisation simultanée des deux lignes de crêtes permet de représenter aussi bien la partie supérieure que la partie inférieure de la surface.

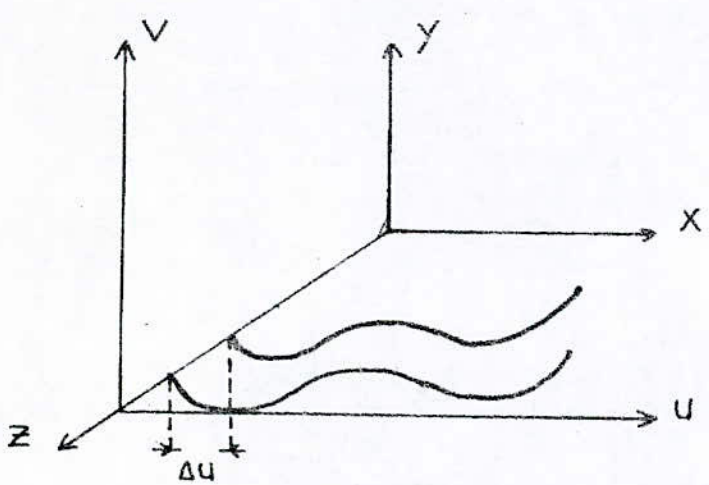

Ligne en cours d'étude
Ligne de crête maximale
Ligne de crête minimale

- figure 3 -

Du fait de la résolution finie du matériel graphique, une ligne est tracée avec un nombre fini de points.
Les valeurs de chaque ligne de crêtes peuvent être rangées dans un vecteur dont la dimension est égale au nombre de points constituant la Courbe.
Ces vecteurs sont initialisés comme suit :

- la ligne de crêtes maximale à la plus petite valeur possible.
- la ligne de crêtes minimale à la plus grande valeur possible.

En passant d'un plan à un autre, du fait d'utiliser une perspective cavalière pour recréer l'illusion du volume, les Courbes sont déplacées d'une distance ΔU ainsi que le montre la figure 4. Il en est de même pour chaque vecteur "ligne de crêtes".

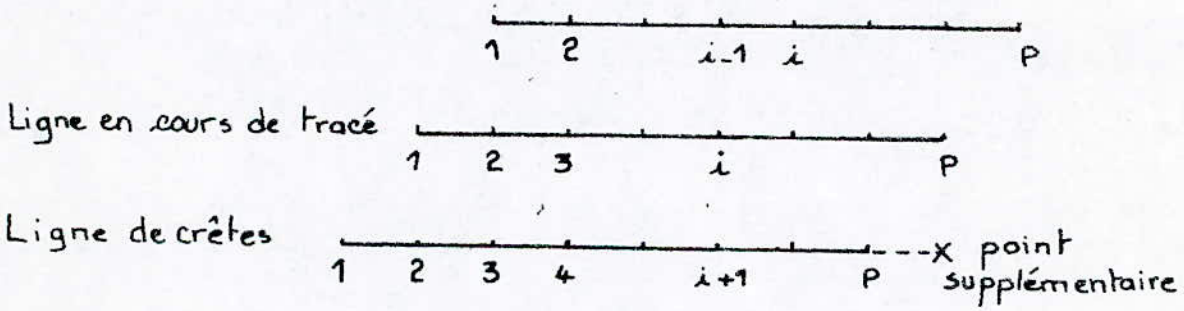


- figure 4 -

L'angle de visée A déterminant le sens du décalage, deux cas se présentent :

2.1. $0 \leq A < 90°$

Les courbes sont décalées vers la droite (fig 4).
Au moment d'effectuer le test des lignes cachées, on doit comparer l'ordonnée de chaque point de rang i à la valeur correspondante (point ayant même abscisse dans le plan UV) de la ligne de crêtes, c'est à dire le point de rang $i+1$.



$0 \leq A < 90$

- figure 5 -

Dans cette figure :

- La courbe en cours de tracé est schématisée par une droite.
- La ligne de crêtes est un vecteur $B(i)_{i=1,...P}$ où sont rangées les ordonnées des points les plus hauts (ou les plus bas).

L'éventuelle mise à jour se fait en rangeant l'ordonnée du point visible Pi dans B(i+1).

Ainsi, lorsqu'on finit de tracer la courbe courante, les lignes de crêtes se trouvent mises à jour mais pas aux points convenables. Il est alors nécessaire de les réajuster en translatant chaque vecteur ligne de crêtes B(K) vers la gauche : alors le contenu de B(i+1) deviendra celui de B(i).

En détaillant davantage la figure 5, il apparait clairement que le dernier point de chaque courbe n'a pas de correspondant dans la ligne de crêtes. Afin de tester sa visibilité, il faut ajouter à la ligne de crêtes, un point de rang P+1 auquel il sera comparé.

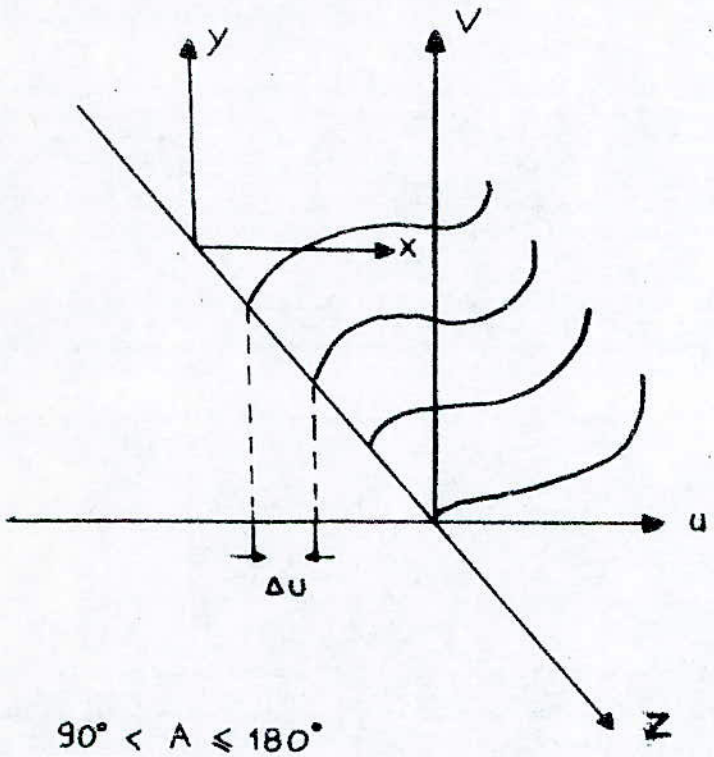
La courbe courante une fois tracée, on réinitialise la ligne de crêtes en ce point à sa valeur maximale ou minimale suivant le cas.

2.2 $90 < A \leq 180$

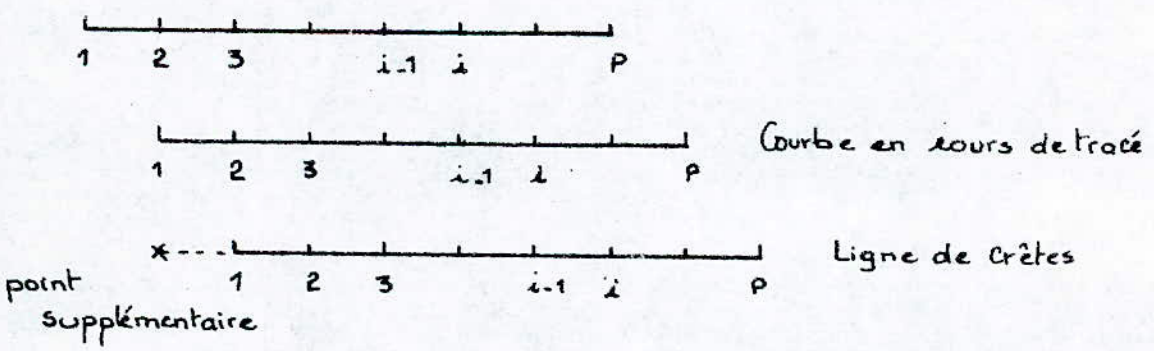
- Les courbes sont décalées vers la gauche (fig. 6).
- Dans ce cas, on compare chaque point de rang i (P_i) au point de rang $i-1$ (P_{i-1}) de la ligne de crêtes dont la dimension est également, $P+1$ (fig. 7).
- L'ordonnée du point P_i est éventuellement rangée dans $B(i-1)$
- Le point supplémentaire se trouve, cette fois, à l'extrême gauche
- Le réajustement des lignes de crêtes se fait en décalant leurs contenus vers la droite : $B(i-1)$ devient $B(i)$.

En résumé

* l'angle A déterminant le rang du point supplémentaire, la dimension des lignes de crêtes doit être égale à $P+2$, P étant le nombre de points constituant la courbe.
* Si $K = Sgn(COS A)$, on compare chaque point P_i de la courbe à tracer au point P_{i+K} de la ligne de crêtes.
* l'éventuelle mise à jour se fait en rangeant l'ordonnée du point P_i dans $B(i+K)$, $B(i)$ étant le vecteur ligne de crêtes.
* En fin, un réajustement des lignes de crêtes s'impose chaque fois qu'une courbe est tracée, sinon, le test des lignes cachées serait erroné.



- figure 6 -



- figure 7 -

2.3 Mise à l'échelle

Référons-nous à la figure 8..a.

Pour tester la visibilité d'un point, on compare son ordonnée à celle du point des lignes de crêtes ayant la même abscisse dans le plan UV.

La distance entre deux points consécutifs de l'axe Z doit, par conséquent, vérifier la relation :

$$\Delta Z * \cos A = \Delta U \qquad 2.2$$

où ΔU représente la distance entre deux points successifs.

Autrement dit, l'abscisse du premier point de chaque courbe est telle que :

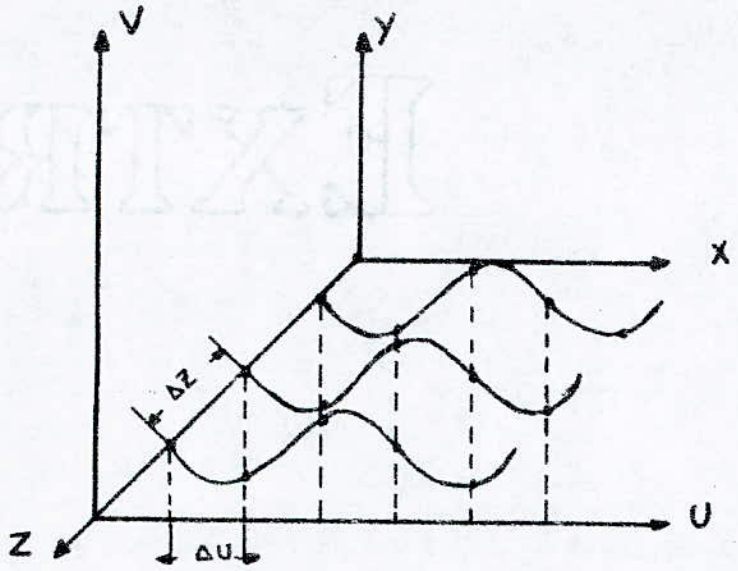
$$U_i = K \Delta U \qquad 2.3$$

où $K \in \mathbb{Z}$

Ainsi, le premier point de chaque courbe se trouvera sur la même verticale que le second point de la courbe précédente.

La distance entre deux points consécutifs d'un même plan étant supposée constante, on aura la coïncidence de tous les points.

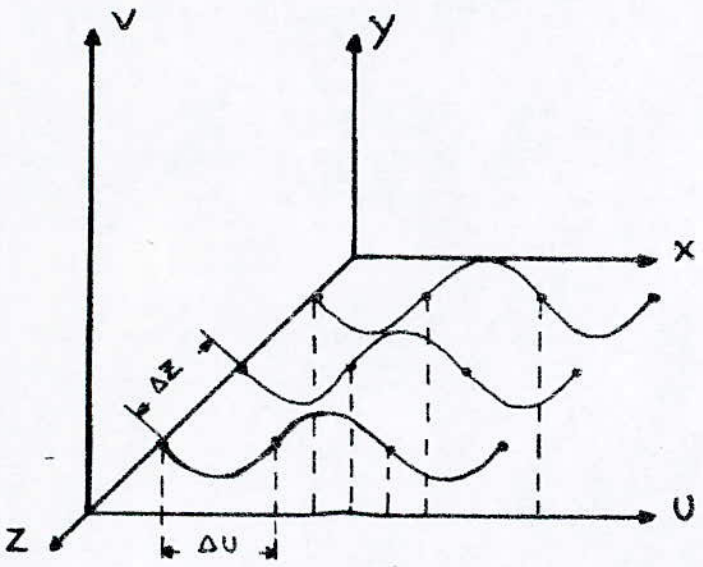
La relation 2.2 peut être réalisée par un choix d'échelle adéquate ; ce qui est toujours possible puisque le problème est de représenter la surface dans le plan UV et non de la calculer.



$\Delta Z * COSA = \Delta U$

Le test des lignes cachées est correct.

- figure 8_a -



$\Delta Z * COSA \neq \Delta U$

Le test des lignes cachées est erroné

- figure 8_b -

Remarques

* En utilisant la résolution maximale de l'écran, la relation 2.2 est implicitement vérifiée. Mais la dimension des lignes de crêtes serait alors prohibitive.

* Des calculs d'interpolation nous permettent également d'avoir la coïncidence des points.

L'ordonnée du point P_i n'ayant pas de correspondant dans la ligne de crêtes, sera déterminée à partir de celles des points P_{i-1} et P_{i+1}.

* Le pas sur U (Δu) étant égal au pas sur X (Δx), La mise à l'échelle qui s'impose pour les coordonnées U et V peut être effectuée sur les coordonnées X, Y, Z

La relation à vérifier est alors :

$$\Delta X = \Delta Z * \cos A \qquad 2.4$$

3 - Cadrage de la surface

Toute surface (ou tout volume dans l'espace) peut-être enfermée dans un parallélépipède dont les dimensions sont fonctions des limites des trois variables X, Y et Z (fig. 9).

Le rectangle de largeur L et de longueur H détermine le plan utilisateur.

Pour visualiser la surface ainsi définie, il est nécessaire de connaitre ses limites puisqu'elles constituent les dimensions de la 'fenêtre' à travers laquelle on voit la surface.

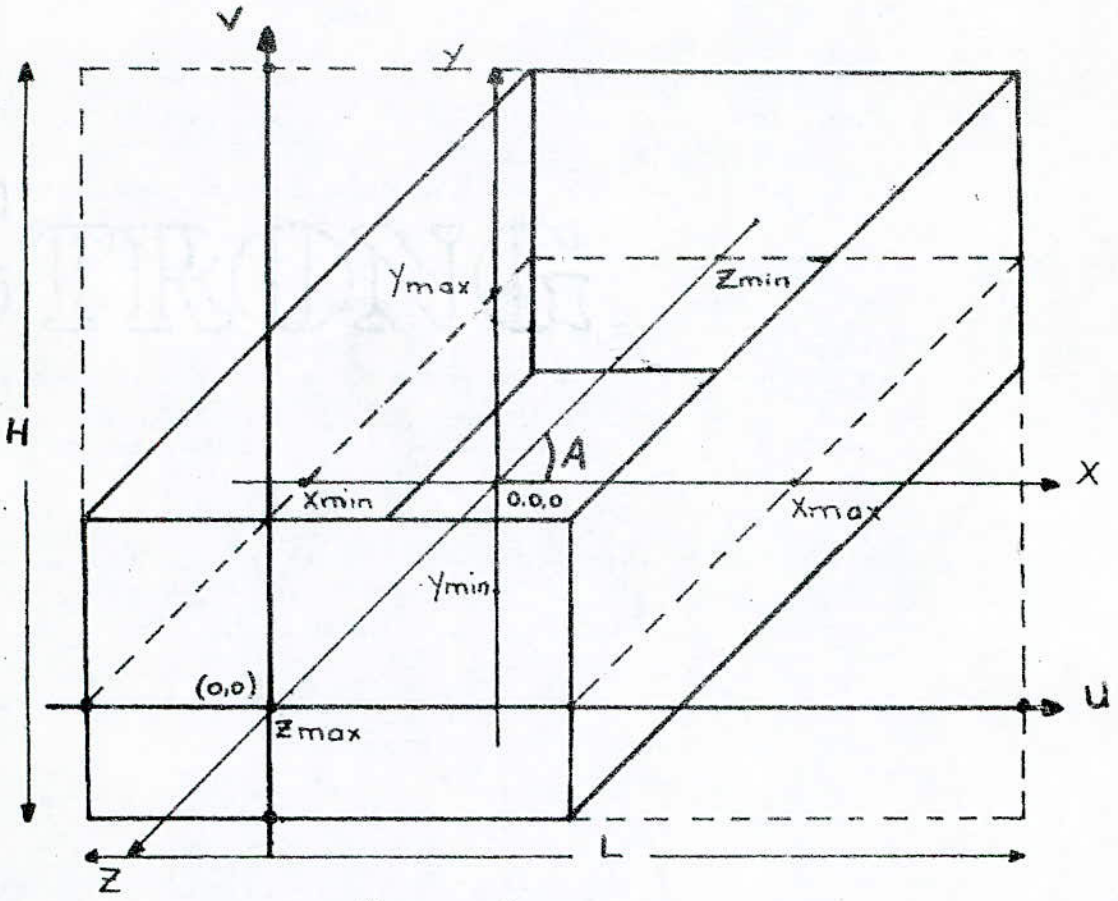
Ces limites se déduisent aisément des limites de cette même surface dans l'espace.

Ainsi, lorsque $0 \leq A \leq \pi/2$ (fig. 9), on a :

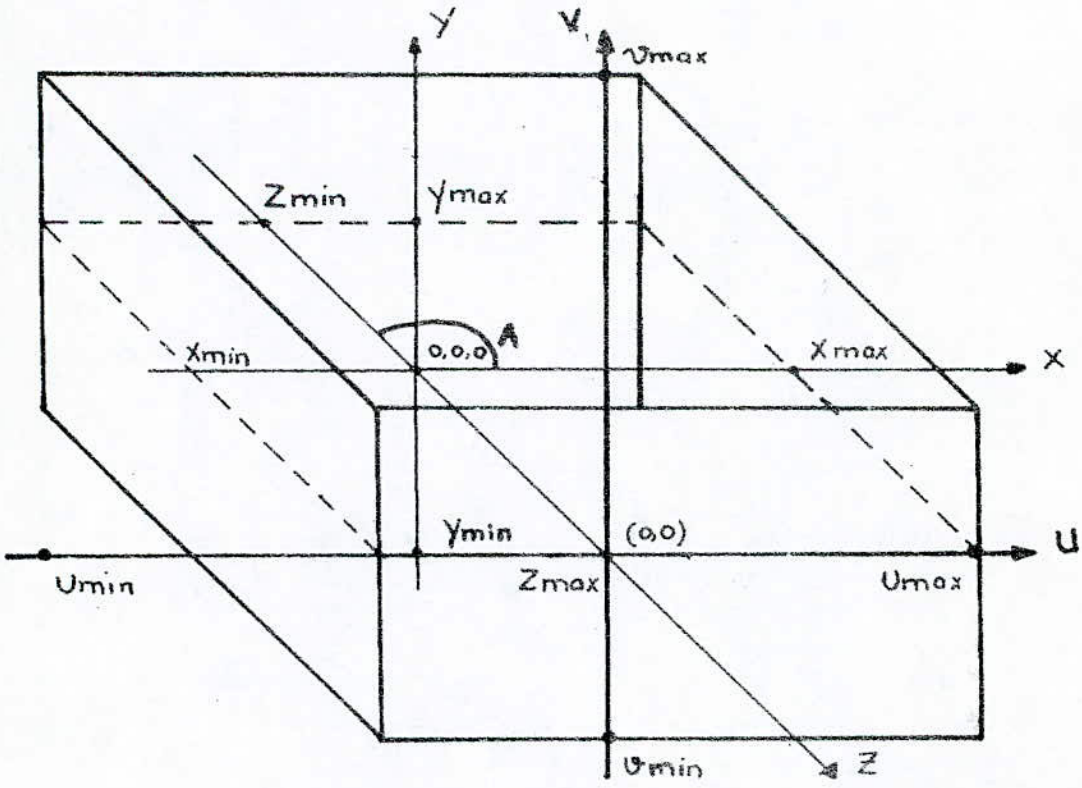
$$\begin{aligned} U_{min} &= X_{min} \\ U_{max} &= X_{max} + (Z_{max} - Z_{min}) * \cos A \\ V_{min} &= Y_{min} \\ V_{max} &= Y_{max} + (Z_{max} - Z_{min}) * \sin A \end{aligned} \qquad 2.5$$

* Lorsque $\pi/2 < A \leq \pi$ (fig. 10), ces équations deviennent :

$$\begin{aligned} U_{min} &= X_{min} + (Z_{max} - Z_{min}) * \cos(\pi - A) \\ U_{max} &= X_{max} \\ V_{min} &= Y_{min} \\ V_{max} &= Y_{max} + (Z_{max} - Z_{min}) * \sin(\pi - A) \end{aligned} \qquad 2.6$$



- figure 9 -



- figure 10 -

4- Transformations de l'affichage.

Aprés avoir visualisé une surface, on peut décider de la modifier.

Ces modifications de l'affichage seront exposées à travers trois transformations de base :

- Translation
- Changement d'échelle.
- Rotation

4-1 Translation

Le déplacement d'un point nécessite seulement un changement de coordonnées correspondant à la translation souhaitée.

Si H, V, P représentent les valeurs des déplacements horizontal, vertical et en profondeur d'un point, la position translatée (XT, YT, ZT) d'un point de coordonnées (X, Y, Z) sera calculée comme suit :

$$XT = X + H$$
$$YT = Y + V \qquad 2\text{-}7$$
$$ZT = Z + P$$

Une valeur positive de H translate un point vers la droite de l'écran ; une valeur positive de V le translate vers le haut et une valeur positive de P le rapproche

alors qu'une valeur négative l'éloigne.
On peut translater des images dans l'espace en appliquant à tous les points de l'affichage les mêmes distances de translation H, V et P.

Si on utilise des valeurs différentes de H, V ou P suivant les points, nous déformons les formes originales de l'affichage. En général, nous avons besoin d'une translation sans déformations, mais de telles distorsions pourront être la base d'expériences de conception d'objets ou être trés utiles pour l'élaboration de jeux.

4-2 Changement d'échelle.

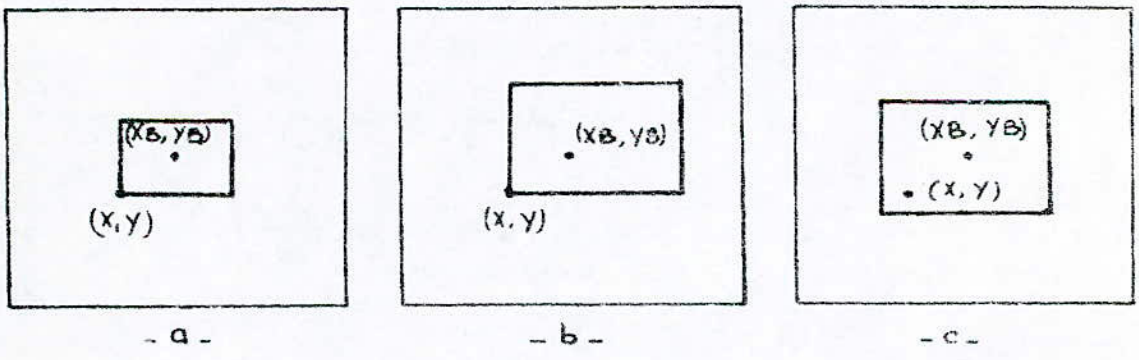
On change les dimensions d'un affichage en multipliant toutes les distances par un coefficient de réduction ou d'agrandissement.
Lorsqu'on fait un changement d'échelle, il faut spécifier la position de l'objet une fois cette opération effectuée.
Par exemple, on peut agrandir un rectangle sans changer la position de son centre, ou sans changer la position du coin inférieur gauche. (fig. 11.)
On doit donc se fixer une position donnée et calculer les nouvelles coordonnées de chaque point de l'écran par rapport à celles-ci.

En choisissant la position de coordonnées (XF, YF, ZF), on peut calculer les nouvelles valeurs (XS, YS, ZS) d'un point de coordonnées initiales (X, Y, Z) par les équations :

$$XS = X * HS + XF * (1 - HS)$$
$$YS = Y * VS + YF * (1 - VS) \qquad 2\text{-}8$$
$$ZS = Z * PS + ZF * (1 - PS)$$

où les rapports d'échelle sont, pour chaque axe : HS, VS et PS.

Les valeurs supérieures à 1 agrandissent l'image alors que les plus petites réduisent la taille de l'affichage.



- figure 11 -

(a) rectangle de centre (XB, YB) et de coin inférieur gauche (X, Y) ;

(b) Un changement d'échelle par rapport au point fixe (X, Y) agrandit la boîte vers le bord supérieur droit de l'écran;

(c) Un changement d'échelle par rapport au point fixe (XB, YB) agrandit la boîte de manière uniforme par rapport au centre

4-3 Rotation

Pour changer l'orientation d'un objet, on calcule, avec les formules suivantes, les coordonnées que devront avoir tous les points de l'affichage aprés rotation.

Dans le cas d'une rotation autour de l'axe Z, de centre (X0, Y0, Z0), les équations de passage sont:

$$XR = X0 + ((X - X0) * \cos A) + ((Y - Y0) * \sin A) \qquad 2.9$$

$$YR = Y0 + ((Y - Y0) * \cos A) - ((Y - Y0) * \sin A)$$

CHAPITRE 3
MISE EN ŒUVRE DE L'ALGORITHME EN BASIC

L'implémentation de l'algorithme sur l'exorciser nous conduirait à écrire en assembleur, en plus du logiciel d'application, les sous-programmes correspondants aux commandes graphiques élémentaires nécessaires à tout tracé (MOVE, DRAW,...), choisir le format des nombres, définir le codage graphique des données, spécifier le mode de transmission etc....
La mise au point d'un tel programme s'avère ardue, les sources d'erreurs étant d'autant plus nombreuses et difficiles à connaître que le programme en question est complexe et que le langage utilisé est davantage "compréhensible" par la machine que par l'opérateur.

Pour toutes les raisons que nous venons d'évoquer, et par souci d'efficacité, nous avons jugé préférable, dans un premier temps, de tester l'algorithme sur un système graphique, en l'occurence, le TEKTRONIX 4051.

3_1. Caractéristiques du TEKTRONIX 4051

Le TEKTRONIX 4051 est un système graphique qui présente un écran rémanent incorporé et ne peut produire que des affichages en noir et blanc.

* La résolution est de 1023 par 780
* La capacité mémoire est de 8 Koctets dont 4K réservés à l'utilisateur.
* Le langage de programmation est le basic.
* Le système comprend une gamme étendue de Commandes spéciales pour graphiques qui lui permettent de travailler en mode "vecteur" ou en mode "point".
* Deux sortes d'unités sont utilisées :
  - les unités graphiques qui s'étendent de :
    0 à 130 sur l'axe des X
    0 à 100 sur l'axe des Y
  - les unités utilisateurs qui peuvent varier de
    - 1.0E+307 à + 1.0E+307.

3.2 - Le programme

Nous ne citerons dans ce paragraphe que les sous-programmes spécifiques au langage évalué utilisé, les autres étant mentionnés en détail au chapitre 4, notamment:

- le rangement des données
- le réajustement des lignes de crêtes.

* Les valeurs de X, Y et Z sont rangées dans deux matrices [P, N]. Les colonnes de rang i de chaque matrice contiennent les valeurs X et Y d'un plan donné.
La valeur de Z pour ce plan est rangée dans une ligne supplémentaire de l'une des deux matrices soit X [P+1, N]

* Le réajustement des lignes de crêtes dépend du signe de COS A, A étant l'angle de visée.
Dans le programme, on pose D1 = Sgn (COSA).
Le décalage se fait dans les deux cas conformément à la relation suivante:

B ((P+3)*(1 - D1) / 2 + D1 * K) =

B((P+3) * (1 - D1) / 2 + D1 * K + D1)

- Ainsi si D1 = 1 $0 \leq A < 90°$ on a:

B (K) = B (K+1) : décalage à droite.

Par contre, si $D1 = -1$ $90° \leq A \leq 180$

ona:

$$B((P+3)-K) = B(P+2-K)$$

Pour le dernier point $K = 1$, on obtient:

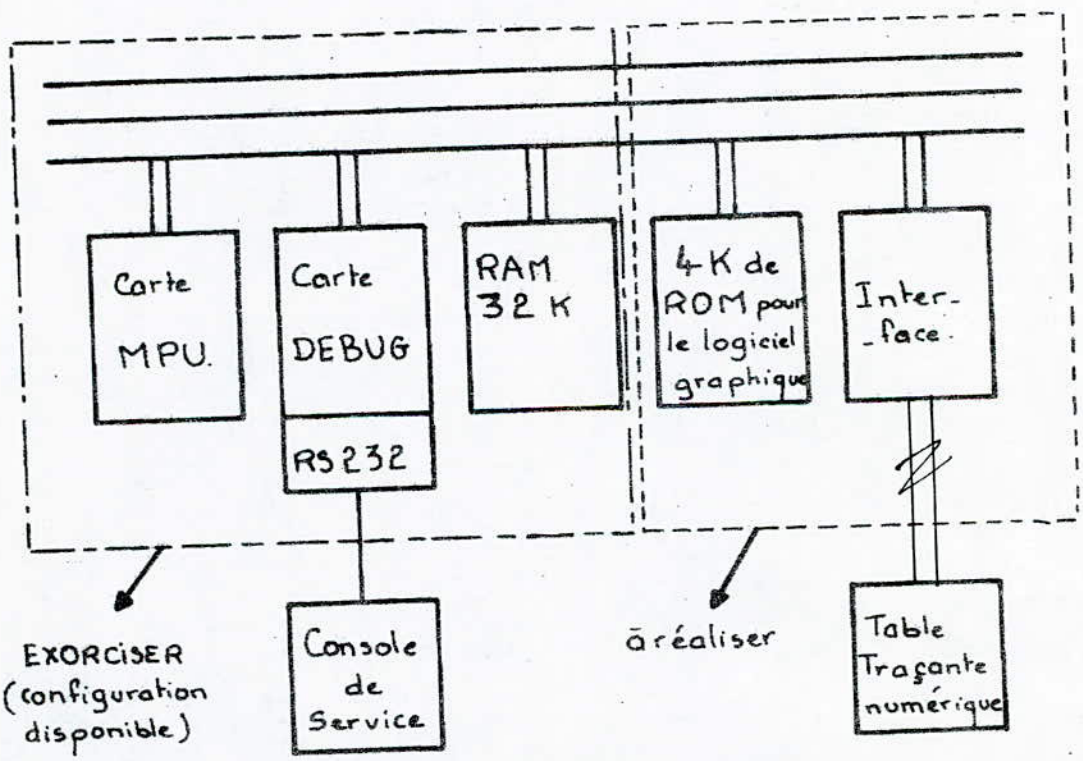
$$B(P+2) = B(P+1)$$

Donc, le décalage se fait vers la gauche mais en commençant par le dernier élément de chaque vecteur ligne de crêtes.

CHAPITRE 4
IMPLEMENTATION SUR L'EXORCISER

1- Configuration du système utilisé

Avec les appareils existants au laboratoire à savoir: EXORCISER, console de service, table traçante numérique, le système minimal requis pour une application graphique peut être représenté conformément à la figure 12



- figure 12 -

Afin de faire de ce système, une structure capable d'évoluer dans le domaine du graphique, il a été nécéssaire

* Du point de vue hardware :

de réaliser une carte d'interface entre l'exorciser et la table traçante TEKTRONIX 4662 et de réserver un emplacement mémoire pour 4 Koctets de ROM destinés à l'implantation définitive du logiciel.
les connexions entre les différents éléments obéis--sent à la norme RS232-C (voir en Annexe l'explication détaillée de cette norme)

* Du point de vue software :

de mettre au point le logiciel nécéssaire au tracé de courbes à trois dimensions et ce, sans disposer d'aucune commande spéciale pour graphiques.
D'où la nécéssité de développer des sous-programmes qui remplissent la fonction de ces dernières.
Il a fallu également choisir la représentation des nombres, et développer des programmes de traitement graphique qui effectue la conversion coordonnées planes → coordonnées table traçante etc....

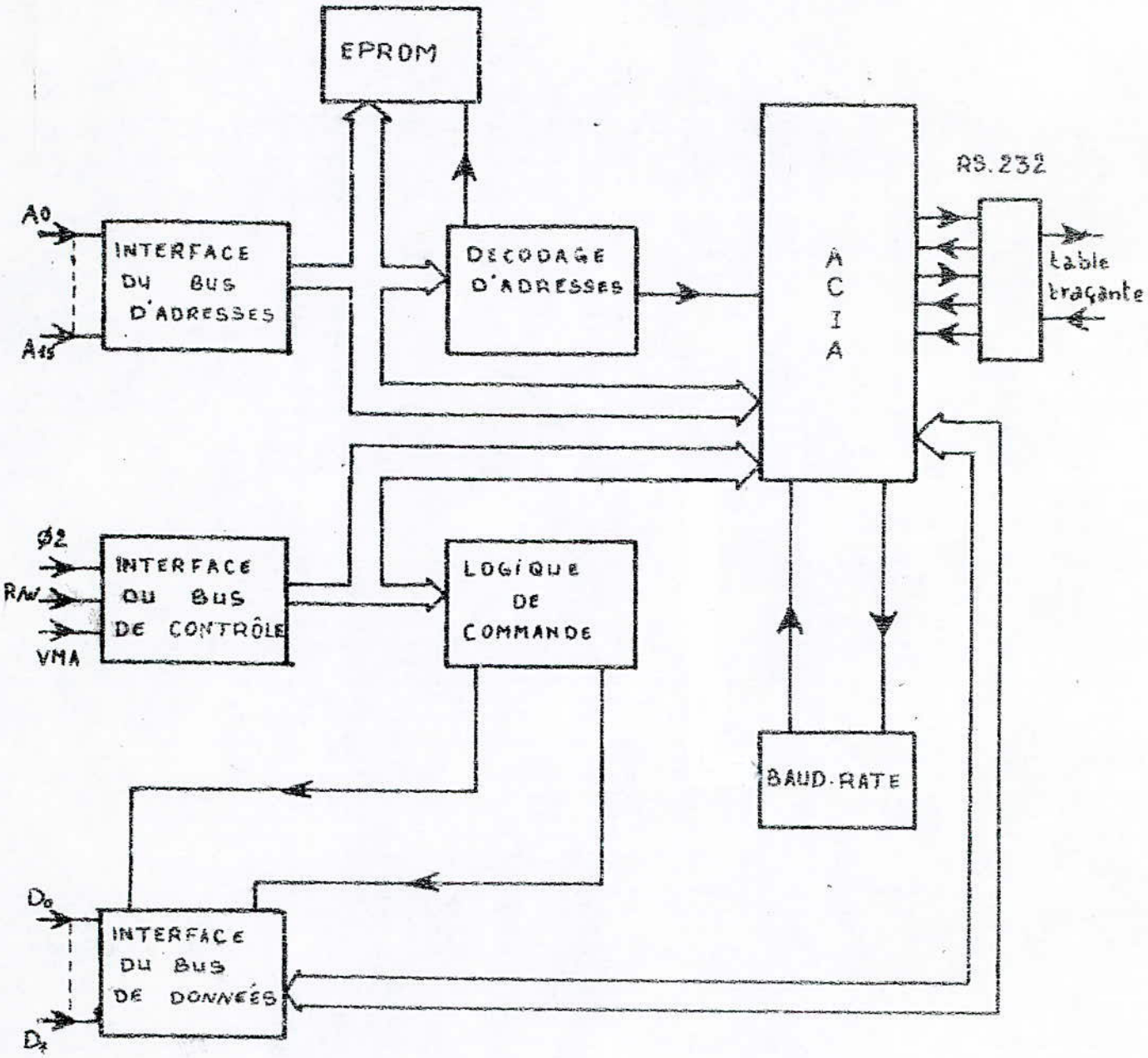
2- Interfaçage : EXORCISER - Table traçante TEKTRONIX 4662

Un microprocesseur ne peut commander directement les périphériques. Un circuit integré d'adaptation appelé interface est nécéssaire entre le microprosseur et le péripherique (à quelques exceptions prés).
Cet interface établira une compatibilité entre les entrées-sorties du processeur et celles du périphérique.
La table traçante est, dans notre cas, utilisée en mode série et obéit à la norme RS-232, d'où l'utilisation de l'ACIA MC 6850 comme circuit d'entrées/sorties.

2-1 Synoptique de la carte d'interface.

La carte comporte essentiellement :

* L'ACIA MC 6850
* Des buffers
* Des circuits de décodage d'adresses
* Un générateur de Baud-rate.
* Une logique de commande.
* 2 EPROM EF 2716



Synoptique de la carte d'interface

- figure 13 -

2-2 Réalisation de la carte

2-2-1 Les circuits d'interface des différents bus

Tous les signaux présents sur les bus du système microprocesseur sont amenés à piloter plusieurs boitiers, d'où la nécessité de les amplifier.
Pour ce faire, on utilise des buffers (8T26) qui, selon l'état des broches (1) et (15), permettent soit la lecture, soit l'écriture. Ils ont, en outre, le rôle d'isoler le bus lorsqu'il est en état de haute impédance.

Les interfaces des bus d'adresses et de Contrôle sont constitués de 8T26 utilisés en mode bidirectionnels.

L'interface du bus de données, lui, comporte des 8T26 en mode unidirectionnel.
Le tableau suivant résume le fonctionnement des 8T26.

ETATS	Broche 1	Broche 15
Lecture	1	1
Ecriture	0	0
Haute impédance	1	0

- Tabl. 1 -

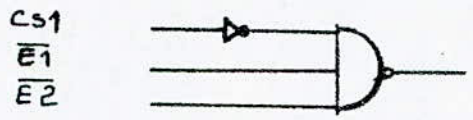
2-2-2 La logique de contrôle des buffers de données

Cette logique de contrôle est appliquée aux broches (1) et (15) des 8T26 ; elle dépend des signaux CS', R/$\overline{W}$, et Ø2 où CS' est la combinaison des signaux :

CS1 : sortie du décodeur d'adresses de l'ACIA

$\overline{E1}$ et $\overline{E2}$: sorties des circuits de décodage d'adresses des 2 EPROM.

Le signal CS doit obéir à la logique câblée suivante:



$CS' = CS1 + E1 + E2$

Un niveau haut sur l'une des entrées : CS1, E1, E2 entraine un niveau haut à la sortie CS. Ainsi, que l'ACIA soit adressé ou l'une des deux EPROM, la logique de Commande dépendra de l'état des broches (1) et (15) des 8T26, de l'horloge extérieure Ø2 et du signal R/$\overline{W}$. Cette logique doit obéir à la table de vérité suivante:

CS'	R/$\overline{W}$	Ø2	Broche 1	Broche 15	Etats des 8T26
1	0	1	0	0	Ecriture
1	1	1	1	1	Lecture
1	0	0	1	0	Haute imp.
1	1	0	1	0	Haute imp.

le schéma fonctionnel est donné à la figure 14.

2.2.3 Décodage d'adresses de l'ACIA.

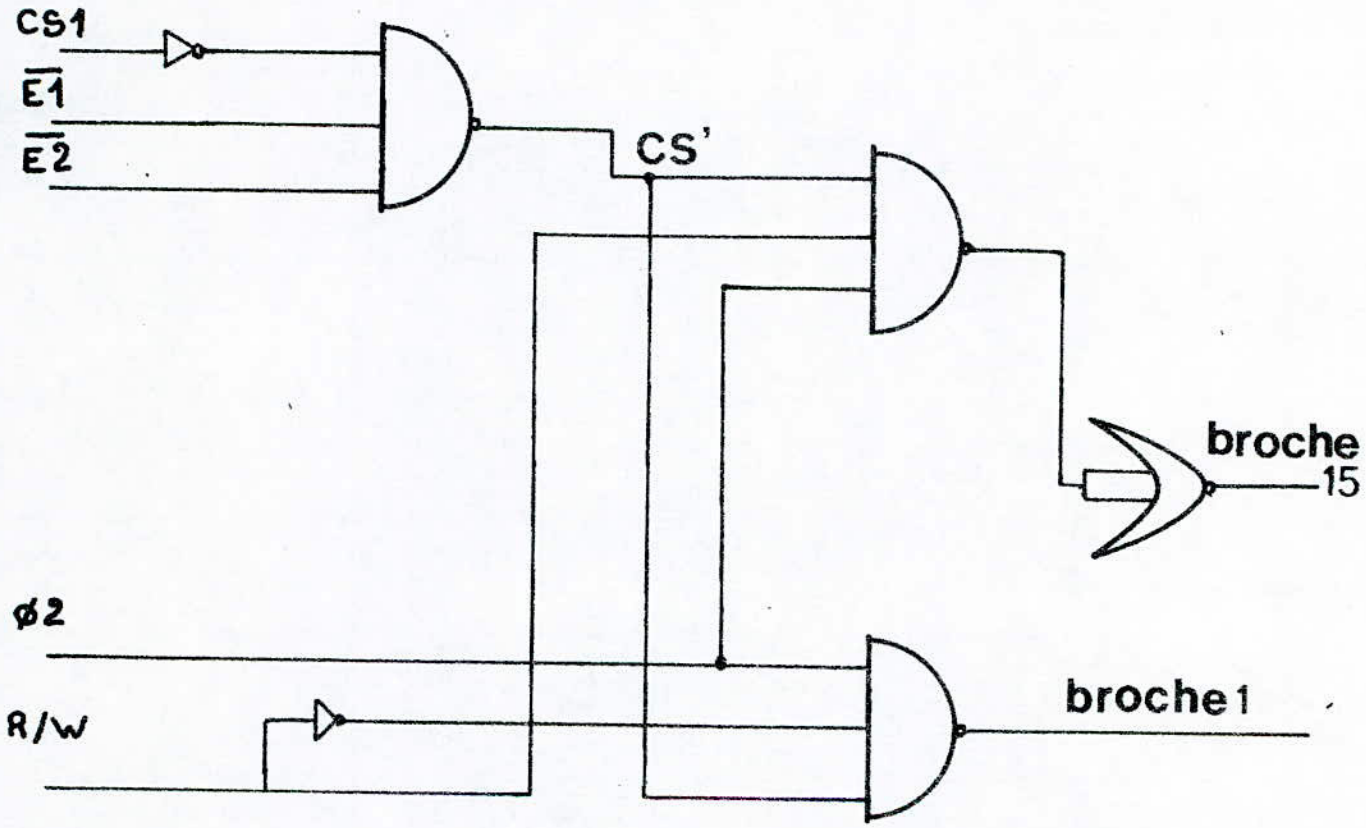
L'ACIA étant sélectionné par une combinaison 1,1,0 des lignes CS0, CS1 et $\overline{CS2}$, on relie :

- CS0 à la ligne VMA
- $\overline{CS2}$ à la masse
- CS1 à la sortie du décodeur d'adresses.

Le choix des positions mémoires D7F0 et D7F1 pour adresser l'ACIA a conduit au circuit de décodage de la figure 15.

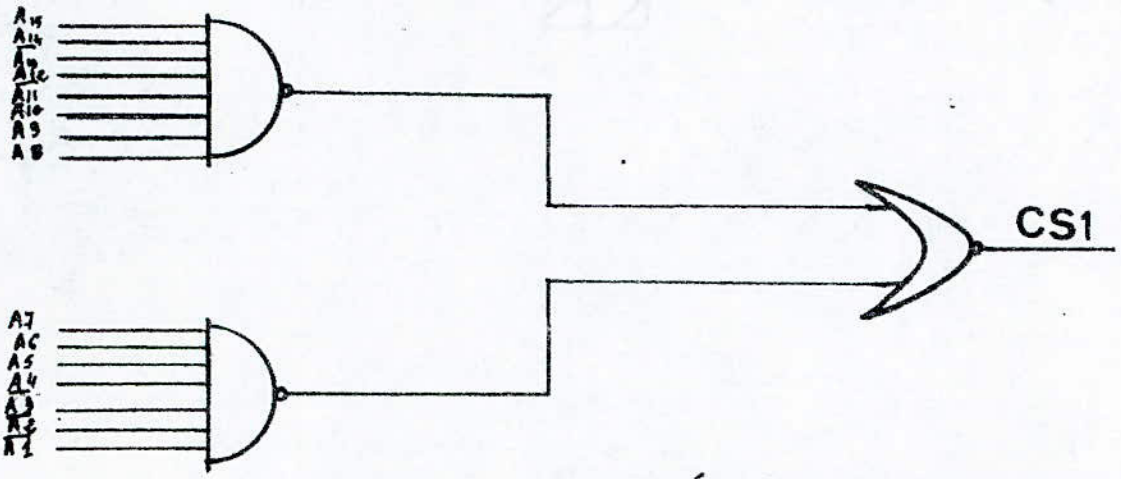
2-2-4 Décodage d'adresses des EPROM

Nous avons attribué à ces EPROM le champ de mémoire s'étalant de: D800 à DFFF pour la première, et de E000 à E7FF pour la seconde.
les 5 bits de poids fort étant fixes, pour chaque EPROM, on décode les 11 bits restants.
Le circuit réalisant ce décodage est réalisé comme l'indique la figure 16.



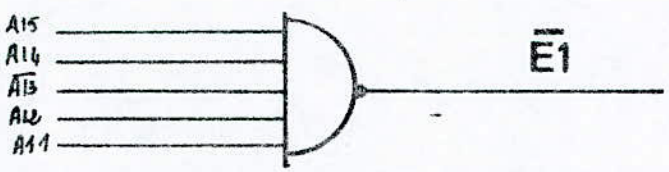
LOGIQUE DE CONTROLE DES BUFFERS des données

- figure 14 -



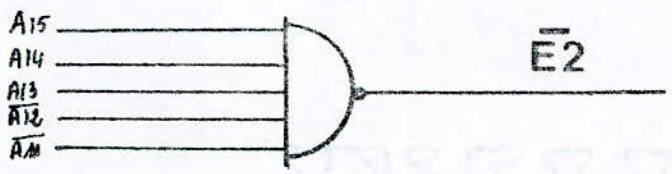
Décodage d'adresses de l'ACIA

- figure 15 -



- figure 16-a -

Décodage d'adresses des EPROM



- figure 16-b -

3 - Logiciel

lors de l'élaboration d'un programme interactif de tracé de courbes 3-D en langage assembleur, le principal problème qui se pose est celui de la représentation des nombres, que ce soit pour les introduire, ou pour les calculs.

3.1 - Représentation des nombres

En raison de sa souplesse et de sa précision, nous avons adopté la représentation en virgule flottante.
Il reste à choisir le format des nombres.
Ce format a, enfait, été imposé par l'utilisation d'une bibliothèque mathématique MOTOROLA qui effectue les différentes opérations arithmétiques en virgule flottante.
Un nombre quelconque positif, négatif, décimal ou entier est écrit en virgule flottante sur 24 bits, soit 16 bits pour la mantisse et 8 bits pour l'exposant.
les bits de poids fort de la mantisse et de l'exposant représentent respectivement le signe du nombre et celui de l'exposant.
En mémoire, le format du chiffre est disposé en :

offset :		
	0	MSByte de la mantisse
	1	LSByte de la mantisse
	2	EXPOSANT

3.2 Organisation de la mémoire

Pour avoir une bonne définition des surfaces visualisées, nous avons prévu :

* un nombre maximum de courbes égal à 35
* un nombre maximum de points par courbe égal à 100.

L'organisation de cette table des données est illustrée par la figure suivante :

1er plan		2ème plan		. . .	35 ième plan	
Adresses	Variables	Adresses	Variables	. . .	Adresses	Variables
$ 100	Z1,	$ 35B	Z 2,	. . .	$ 5116	Z35.
$ 103	X1	$ 35E	X1	. . .	$ 5119	X1.
$ 106	Y1	$ 361	Y1	. . .	$ 511C	Y1
.	.	.	.	. . .	.	.
.	.	.	.	. . .	.	.
$ 355	X100	$ 5B0	X100	. . .	$ 536B	X100
$ 358	Y100	$ 5B3	Y100	. . .	$ 536E	Y100

Tabl. 2

3.3 - Développement du logiciel

Comme il a été dit au paragraphe 2 du chapitre 2, le programme de tracé d'une courbe à 3 dimensions peut se décomposer en plusieurs parties :

Certaines sont essentielles comme :

- le stockage des données
- la mise à l'échelle (pour le traçé)
- le calcul de U et V et le test des lignes cachées.

D'autres sont auxilliaires comme:

- le cadrage de la courbe
- la recherche des extrèma de X, Y et Z

3.3.1 Stockage des données

Le but de ce programme est essentiellement de représenter des surfaces dont les paramètres proviennent de tableaux d'acquisition de mesures. Le stockage des données est de ce fait, nécéssaire.

Son principe est le suivant : dans la mémoire, les paramè-tres sont rangés dans un vecteur dont le premier élément est la valeur de Z correspondant au plan le plus proche de l'observateur. Suivent ensuite les coordonnées des points appartenant à ce plan, puis la valeur de Z suivante etc....

* Surface décrite par une fonction de deux variables

Lorsque la surface est décrite par une fonction $y = f(x, z)$, les données à introduire sont respectivement Zmin, Zmax, Xmin, Xmax, le nombre de courbes (N), le nombre de points par courbe (P), COSA puis SINA A étant l'angle de Visée.

Ces données sont acquises en décimal puis converties en virgule flottante avant d'être rangées dans les mémoires réservées à cet effet.

Leur traitement commencera par le calcul du pas sur X (ΔX) et du pas sur Z (ΔZ) ainsi définis :

$$\Delta X = \frac{X_{max} - X_{min}}{P - 1} \qquad 3.1$$

$$\Delta Z = \frac{Z_{max} - Z_{min}}{N - 1} \qquad 3.2$$

La première valeur de Z, c'est à dire Zmax, sera ensuite transférée dans la mémoire $100, suivie de Xmin et de la valeur correspondante de y que l'on aura calculée. On calculera également les coordonnées (X_i, Y_i) du point suivant de ce même plan par les équations :

$$X_i = X_{i-1} + \Delta X$$
$$Y_i = f_z(X_i) \qquad 3.3$$

avant de les ranger à leur tour.

On continue le processus de calcul puis de transfert jusqu'à ce que tous les points de tous les plans aient été stockés.

* Surface décrite par des tableaux de mesures

Dans ce cas, toutes les données étant disponibles dans des tableaux, il ne reste plus qu'à les stocker.

Les couples (X, Y) de chaque tableau doivent être rangés suivant les X croissants, chaque abscisse X étant immédiatement suivie de l'ordonnée Y correspondante.

Les tableaux doivent être classés pour être introduits suivant les Z décroissants.

On commencera alors par introduire respectivement : le nombre N de tableaux, le nombre P de points par tableau, COSA puis SINA, le pas sur X (ΔX) et enfin le pas sur Z (ΔZ).

Suivront ensuite les données introduites dans l'ordre suivant :

$Z1, X1, Y1, \ldots\ldots X_P, Y_P, Z2, \ldots Z_N, X1, Y1, \ldots X_N, Y_N$

3-3-2 Mise à l'échelle

Cette routine permet de calculer les nouvelles valeurs de X qui vérifient la relation

$$\Delta X' = \Delta Z * \cos A$$

$\Delta X'$ étant le nouveau pas sur X

Ces valeurs se déduisent des anciennes par la relation:

$$X'_i = X_i * \frac{\Delta Z * \cos A}{\Delta X} \qquad 3-4$$

et les remplacent en mémoire.

3-3-3 Cadrage de la surface

Pour représenter la fonction, il est nécessaire de spécifier les dimensions de la fenêtre à travers laquelle elle est vue.

L'instruction Basic "Window" est remplacée dans le cas, par un sous programme de changement d'échelle qui permet d'accéder à tous les points adressables de la table traçante à savoir :

- 1024 points selon l'axe des X
- 2732 points selon l'axe des Y

Pour ce faire, la transformation suivante s'impose :

Umin $\longrightarrow$ 0

Umax $\longrightarrow$ $(1023)_{10}$

d'où : $Umax - Umin = (1023)_{10}$

Pour que le zéro soit l'origine, on doit donc retrancher à chaque valeur (ui) la quantité Umin. Une simple règle de trois nous permet alors, de calculer la nouvelle coordonnée Ui à envoyer, c'est à dire NUi.

$$NUi = \frac{(Ui - Umin) * (1023)_{10}}{Umax - Umin} \qquad 3-5$$

Un raisonnement analogue permet d'écrire la transformation à effectuer suivant l'axe vertical Y :

$$NVi = \frac{(Vi - Vmin) * (2731)_{10}}{VMAX - VMIN} \qquad 3-6$$

En résumé, afin de visualiser entièrement la surface, il suffit, connaissant Umin, Umax, Vmin, Vmax, de calculer les coordonnées du point à tracer puis d'effectuer le changement d'échelle mentionné ci-dessus.

3-3-4 Recherche des extrèma de X, Y et Z

Cette recherche est imposée par la nécessité de connaître les limites de la surface considérée dans le plan de représentation.

On recherchera d'abord les extrèma de Z puis de X et enfin de Y par une même routine dont le principe est

le suivant :

On initialise le minimum et le maximum à la première valeur de la variable considérée puis on compare la valeur suivante au minimum et éventuellement au maximum jusqu'à l'obtention des extrêma.

3-3-5 Tracé

Ce programme commence par une routine d'initialisation de :

- la ligne de crêtes maximale à BFFF10 qui est la plus petite valeur possible vu le format des nombres.

- la ligne de crêtes minimale à 7FFF10 qui est la plus grande valeur possible.

Les éléments de chaque ligne de crêtes sont rangés dans une zone mémoire dont l'adresse de début dépend de l'angle de visée A (ref paragraphe 2.1, chap 2) Ainsi, le premier point de la ligne de crêtes minimale sera rangé à l'adresse $5372 si $A > 90°$ alors qu'il serait rangé en $5378 si $0 \leq A < 90°$.

Cette routine est suivie d'un sous-programme de calcul des coordonnées U et V puis du test des lignes cachées.

De plus, Comme il a été dit au chapitre 2, la visibilité d'un point est testée en comparant son ordonnée à celle du point correspondant des lignes de crêtes.
l'éventuelle mise à jour de ces dernières se fait par une routine de transfert des données.
Lorsque tous les points d'une courbe sont traités, on réajuste les lignes de crêtes en décalant le contenu des mémoires qui leurs sont allouées dans un sens ou dans l'autre suivant la valeur de l'angle de visée.
Cependant, lorsque les coordonnées U et V d'un point sont calculées, que ce point soit visible ou non visible, un même traitement doit lui être appliqué afin de l'envoyer à la table traçante à savoir :

* le codage graphique des données.
* le sous programme MOVE ou DRAW

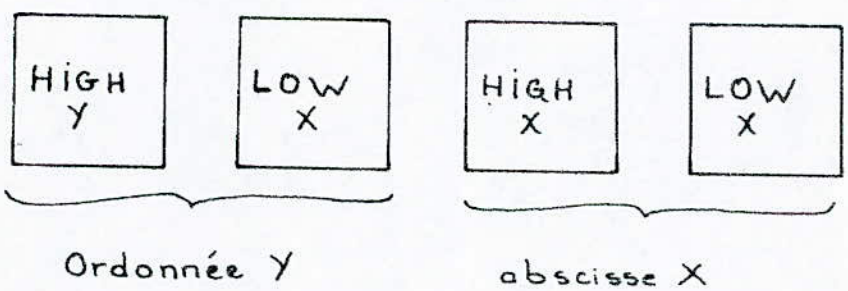
* <u>le codage graphique des données</u>.

Les données sont transmises sous forme de caractères codés en ASCII sur 7 bits. Au moyen des caractères ASCII US et GS, la table traçante TEKTRONIX travaille respectivement en mode alphanumérique et en mode graphique.
Dans le premier cas, le traceur imprime sur la table

les caractères ASCII transmis.
Dans le second, ces derniers ne sont plus imprimés tels qu'ils sont reçus, mais sont interprétés comme les coordonnées d'un point de la table traçante.
Ce mode est davantage sollicité puisque le but de notre travail est de visualiser des courbes.

Chaque point de la table traçante est représenté par une paire de coordonnées (Y, X) codées en 4 caractères ASCII : deux pour l'ordonnée Y et deux pour l'abscisse X. La séquence des caractères ASCII prend la forme suivante

Une paire de coordonnées



- figure 17 -

Ces deux coordonnées sont décodées par une logique interne pour déterminer la position sur la table du point à représenter en leurs affectant des codes appropriés.

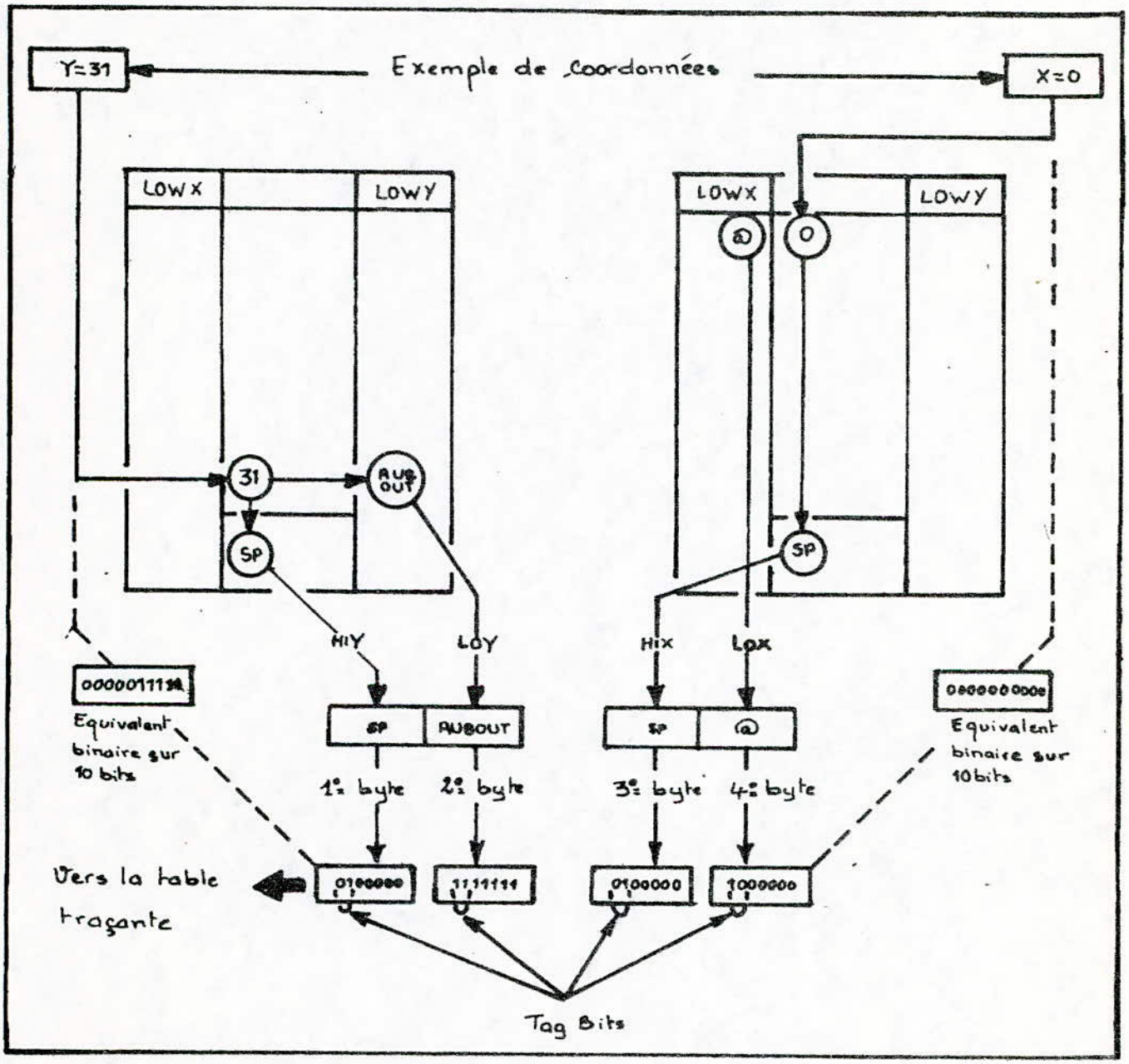
Remarquons qu'un extra-byte (XLOY) est parfois utilisé pour avoir une meilleure résolution (4096 points sur l'axe x).
Le tableau suivant résume les codes des différents bytes.

High Y	01
Low Y	11
High X	01
Low X	10

- Tabl. 3 -

Il existe une relation directe entre la valeur décimale d'une coordonnée, son équivalent binaire et les caractères ASCII utilisées pour la représenter.
Le principe du codage est le suivant:
On écrit l'équivalent binaire de la coordonnée décimale. Ce nombre de 10 bits est ensuite décomposé en deux nombres binaires de 5 bits chacun. Les caractères ASCII correspondants sont obtenus par l'adjonction de deux bits de poids fort à chacun d'eux.
On ajoutera, par exemple 01 pour High Y ....

La figure ci-dessous illustre la manière dont les quatres caractères ASCII : SPACE, RUBOUT, SPACE, @ sont interpretés comme la paire de coordonnées (31, 0)



- figure 18 -

* Sous programmes MOVE et DRAW

Lorsqu'un point est visible, on doit tracer le segment qui le relie au point précédent; lorsqu'il est non visible, le curseur doit se positionner en ce point sans tracer de ligne. Ces deux opérations sont effectuées en BASIC au moyen des instructions MOVE et DRAW.

ELLes sont remplacées dans ce cas par des sous programmes d'émission de caractères

Dans les deux cas; on doit se placer en mode graphique par l'envoi du caractère ASCII GS (graph submode) que l'on fait suivre :

- de la séquence des quatres caractères ASCII formant la paire de coordonnées du point si on doit faire un " MOVE ".

- du caractère ASCII `BEL' puis de la paire de coordonnées si on doit tracer le segment reliant ce point au précédent (DRAW)

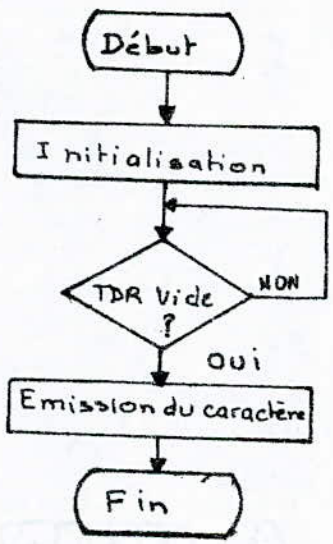
On distinguera donc les deux sous-programmes par le nombre de caractères émis à l'ACIA que l'on devra initialiser au préalable puisque la transmission se fait en mode série asynchrone.

* Le sous-programme d'initialisation de l'ACIA permet de faire une remise à zéro du circuit et de choisir le mode de fonctionnement.

Soit ACIAS l'adresse du registre d'état et de contrôle ; en choisissant le mode divisé par 16, une transmission avec 8 bits, pas de parité et 2 bits de stop, on écrira :

```
LDA  A  #$03
STA  A  ACIAS      : reset de l'ACIA
LDA  A  #$11
STA  A  ACIAS      : Configuration de
                     l'ACIA
RTS
```

* Aprés l'initialisation de l'ACIA, l'émission d'un caractère ne peut se faire que si le registre de transmission est vide. Ce qui est montré par l'organigramme ci-joint.



Le sous-programme d'émission d'un caractère est le suivant:

```
LDA B   ACIAS
AND B   # $02      Ne laisse que le bit B1 dans B
BEQ     EMICAR     registre plein.
STA A   ACIAS      Donnée à émettre dans TDR
RTS
```

Remarquons que le caractère à transmettre se trouve, dans le cas de ce programme, dans l'accumulateur A

En résumé, une fois les coordonnées U et V calculées en virgule flottante, les opérations suivantes sont effectuées:

* Conversion de U et V en des coordonnées comprises entre 0 et 1023 (U) et 0 et 2731 (V) en virgule flottante. C'est le rôle du sous programme "Window"

* Les valeurs de U et V ainsi obtenues sont converties en hexadecimal sur 16 bits.

* On effectue alors le codage graphique des données

* Puis, suivant le cas, on appelle le sous-programme MOVE ou DRAW.

Ces sous-programmes sont donnés ci-joint

```
                        *
         56A0     A MCC     EQU   $56A0
         56A1     A DLIST   EQU   MCC+1
                        *
E0F6 86 05        A MOVE    LDAA  #5
E0F8 B7 56A0      A         STAA  MCC
E0FB CE 56A1      A         LDX   #DLIST
E0FE 86 1D        A         LDAA  ##$1D
E100 D6 46        A         LDAB  AY
E102 A7 00        A         STAA  0,X
E104 E7 01        A         STAB  1,X
E106 96 47        A         LDAA  AY+1
E108 D6 43        A         LDAB  AX
E10A A7 02        A         STAA  2,X
E10C E7 03        A         STAB  3,X
E10E 96 44        A         LDAA  AX+1
E110 A7 04        A         STAA  4,X
E112 BD E13A      A         JSR   ACIA
E115 39                     RTS
                        *
                        *
                        *
E116 86 06        A DRAW    LDAA  #6
E118 B7 56A0      A         STAA  MCC
E11B CE 56A1      A         LDX   #DLIST
E11E 86 1D        A         LDAA  #$1D
E120 C6 07        A         LDAB  #$07
E122 A7 00        A         STAA  0,X
E124 E7 01        A         STAB  1,X
E126 96 46        A         LDAA  AY
E128 D6 47        A         LDAB  AY+1
E12A A7 02        A         STAA  2,X
E12C E7 03        A         STAB  3,X
E12E 96 43        A         LDAA  AX
E130 D6 44        A         LDAB  AX+1
E132 A7 04        A         STAA  4,X
E134 E7 05        A         STAB  5,X
E136 BD E13A      A         JSR   ACIA
E139 39                     RTS
                        *
                        *
                        *
                        *
                        *
E13A 86 03        A ACIA    LDAA  #3
E13C B7 D7F0      A         STAA  ACIS
E13F 86 11        A         LDAA  #$11
E141 B7 D7F0      A         STAA  ACIS
E144 F6 56A0      A         LDAB  MCC
E147 CE 56A1      A         LDX   #DLIST
E14A A6 00        A ACIA0   LDAA  0,X
E14C BD E154      A         JSR   ACIA1
```

Initialisation du nombre de caractères à emettre

Stockage du caractère ASCII 'GS'

Stockage des coordonnees Hexadécimales U et V

Emission des caractères

Initialisation du nombre de caractères à emettre.

Stockage des caractères ASCII 'GS' et 'BEL'

Stockage des coordonnées Hexadécimales U et V

Emission des caractères

Reset de l'ACIA

Configuration de l'ACIA

Pointer l'adresse de début de liste

```
GRAPHE  .SA:0  GRAPH1

A E14F 08                       INX
A E150 5A                       DECB
A E151 26 F7 E14A               BNE     ACIA0
A E153 39                       RTS
                        *
                        *
A E154 F7 569F   A ACIA1  STAB   TAMPON
A E157 F6 D7F0   A EMICAR LDAB   ACIS
A E15A C4 02     A        ANDB   #2
A E15C 27 F9 E157        BEQ    EMICAR
A E15E B7 D7F1   A       STAA   ACID
A E161 F6 569F   A       LDAB   TAMPON
A E164 39                RTS
                        *
                        *
          D7F0   A ACIS   EQU    $D7F0
          D7F1   A ACID   EQU    ACIS+1
          569F   A TAMPON EQU    $569F
                        *
                        *       ***************************************************
                        *       *                    S/P: WINDOW                  *
                        *       *   CONVERTIT LES Ui(resp:Vi)EN DES NOMBRES      *
                        *       *   COMPRIS ENTRE 0 ET 1023 (resp:0 ET 2731)     *
                        *       ***************************************************
          0070   A MCAL1  EQU    $70
          0022   A UMIN   EQU    MINU
          0028   A VMIN   EQU    MINV
                        *
E165 86 43       A WINDOW LDAA   #AX
E167 C6 22       A        LDAB   #UMIN
E169 BD E4DA     A        JSR    FPSUB
E16C 86 43       A        LDAA   #AX
E16E C6 A6       A        LDAB   #M1023
E170 BD E5D7     A        JSR    FPMUL
E173 86 70       A        LDAA   #MCAL1
E175 C6 25       A        LDAB   #UMIN+3
E177 BD E4B8     A        JSR    FPCPY
E17A 86 70       A        LDAA   #MCAL1
E17C C6 22       A        LDAB   #UMIN
E17E BD E4DA     A        JSR    FPSUB
E181 86 43       A        LDAA   #AX
E183 C6 70       A        LDAB   #MCAL1
E185 BD E647     A        JSR    FPDIV
E188 86 46       A        LDAA   #AY
E18A C6 28       A        LDAB   #VMIN
E18C BD E4DA     A        JSR    FPSUB
E18F 86 46       A        LDAA   #AY
E191 C6 A9       A        LDAB   #M2731
E193 BD E5D7     A        JSR    FPMUL
E196 86 70       A        LDAA   #MCAL1
```

Ne laisse que le bit b1 dans B
Registre plein
Donnée à émettre dans TDR

Calcul de l'expression

$$NUi = \frac{(Ui - U_{min}) * (1023)_{10}}{U_{max} - U_{min}}$$

```
3A E193 C6 2B     A        LDAB   #VMIN+3
9A E19A BD E4B8   A        JSR    FPCPY
DA E19D 86 70     A        LDAA   #MCAL1
LA E19F C6 28     A        LDAB   #VMIN
2A E1A1 BD E4DA   A        JSR    FPSUB
3A E1A4 86 46     A        LDAA   #AY
~~A E1A6 C6 70    A        LDAB   #MCAL1~~
DA E1A8 BD E647   A        JSR    FPDIV
```

Calcul de l'expression

$$NVi = \frac{(Vi - Vmin) * (2731)_{16}}{Vmax - Vmin}$$

GRAFHE .SA:0 GRAFH1

```
                    *
                    *       CONVERSION FP ===> HEXADECIMAL
                    *       ------------------------------
                    *
A E1AB DE 43      A         LDX    AX
A E1AD 96 45      A         LDAA   AX+2
A E1AF BD E47D    A         JSR    PSH4
A E1B2 30                   TSX
A E1B3 E6 00      A         LDAB   0,X
A E1B5 86 43      A         LDAA   #AX
A E1B7 BD E6F1    A         JSR    FPFIX
A E1BA 31                   INS
A E1BB 31                   INS
A E1BC 31                   INS
A E1BD 31                   INS
                    *
A E1BE DE 46      A         LDX    AY
A E1C0 96 48      A         LDAA   AY+2
A E1C2 BD E47D    A         JSR    PSH4
A E1C5 30                   TSX
A E1C6 E6 00      A         LDAB   0,X
A E1C8 86 46      A         LDAA   #AY
A E1CA BD E6F1    A         JSR    FPFIX
   E1CD 31                  INS
   E1CE 31                  INS
   E1CF 31                  INS
   E1D0 31                  INS
                    *
                    *       CONVERSION HEXADECIMAL-CODE CARTE
                    *       ---------------------------------
                    *
   E1D1 96 46       A        LDAA   AY
   E1D3 84 03       A        ANDA   #3
   E1D5 97 46       A        STAA   AY
   E1D7 78 0047     A        ASL    AY+1
   E1DA 79 0046     A        ROL    AY
```

Mantisse de Ui dans l'index
Exposant de Ui dans Acc A
Sauvegarde dans la pile
Pointer l'adresse du résultat
Conversion en hexadécimal

Mantisse de Vi dans l'index
Exposant de Vi dans Acc A
Sauvegarde dans la pile
Pointer l'adresse du résultat
Conversion en Hexadécimal

Ne laisse que les bits b_1, b_2 dans AY

```
:A E1DD 78 0047  A        ASL    AY+1
:A E1E0 79 0046  A        ROL    AY
:A E1E3 78 0047  A        ASL    AY+1
:A E1E6 79 0046  A        ROL    AY
:A E1E9 74 0047  A        LSR    AY+1
:A E1EC 74 0047  A        LSR    AY+1
:A E1EF 74 0047  A        LSR    AY+1
:A E1F2 96 46    A        LDAA   AY
:A E1F4 8A 20    A        ORAA   #$20
:A E1F6 97 46    A        STAA   AY
:A E1F8 D6 47    A        LDAB   AY+1
:A E1FA CA 60    A        ORAB   #$60
:A E1FC D7 47    A        STAB   AY+1
                 *
:A E1FE 96 43    A        LDAA   AX
:A E200 84 03    A        ANDA   #3
:A E202 97 43    A        STAA   AX
:A E204 78 0044  A        ASL    AX+1
:A E207 79 0043  A        ROL    AX
:A E20A 78 0044  A        ASL    AX+1
:A E20D 79 0043  A        ROL    AX
:A E210 78 0044  A        ASL    AX+1
```

Transfert des 3 bits de poids fort de AY+1 dans AY

Insérer le Code de HiY

Insérer le Code de LOY

Ne laisse que les bits b_1, b_2 dans AX

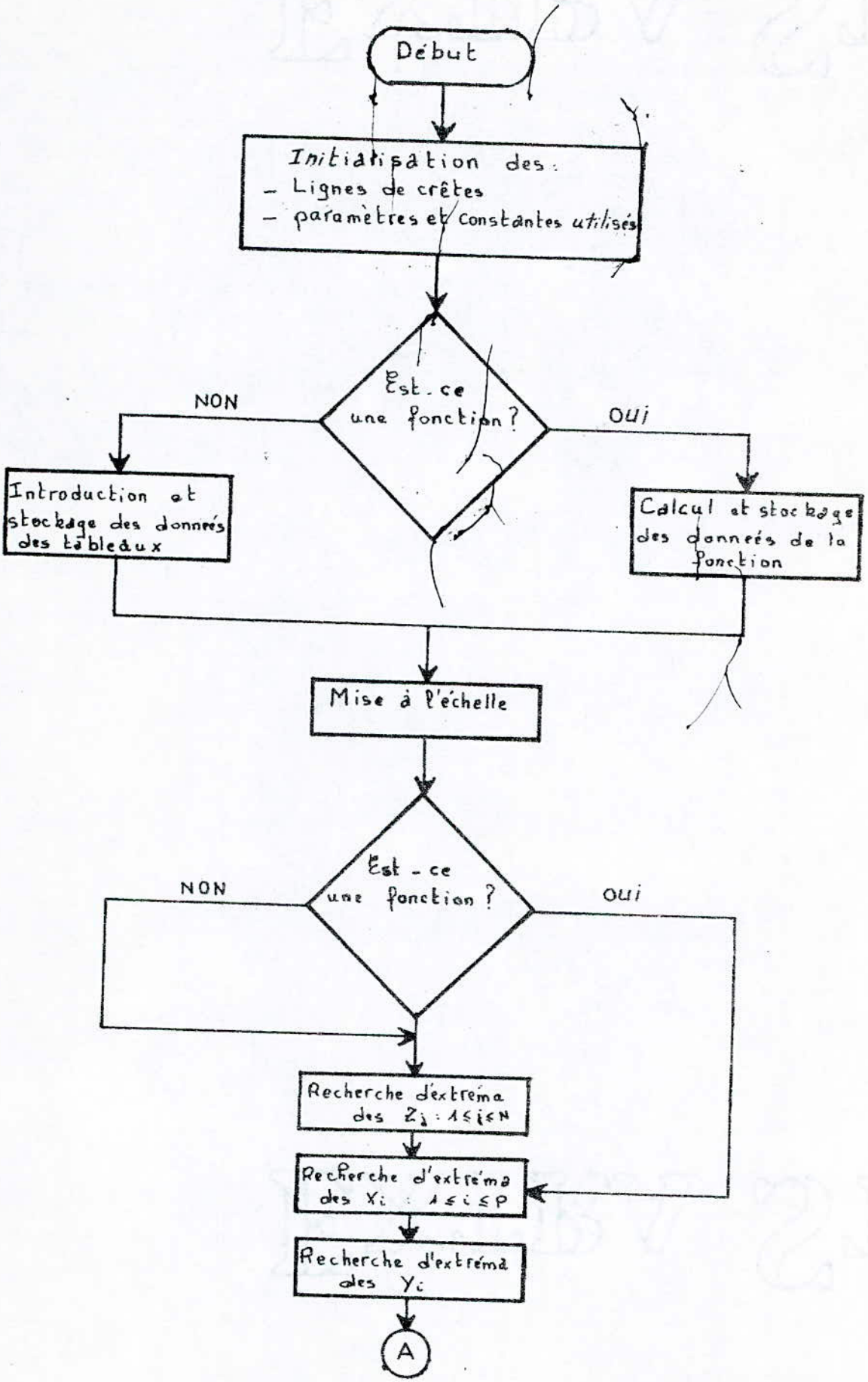
Transfert des 3 bits de poids

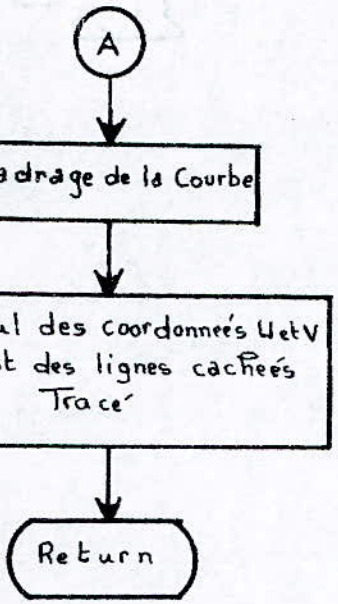
```
GRAPHIC   .SA:0  GRAPHI

:A E213 79 0043  A        ROL    AX
:A E216 74 0044  A        LSR    AX+1
:A E219 74 0044  A        LSR    AX+1
:A E21C 74 0044  A        LSR    AX+1
:A E21F 96 43    A        LDAA   AX
:A E221 8A 20    A        ORAA   #$20
:A E223 97 43    A        STAA   AX
:A E225 D6 44    A        LDAB   AX+1
:A E227 CA 40    A        ORAB   #$40
:A E229 D7 44    A        STAB   AX+1
:A E22B 39               RTS
                 *
                 *
   E22C 7E D800  A TERM   JMP    START
                 *
```

Fort de AX+1 dans AX

Insérer le code de HiX

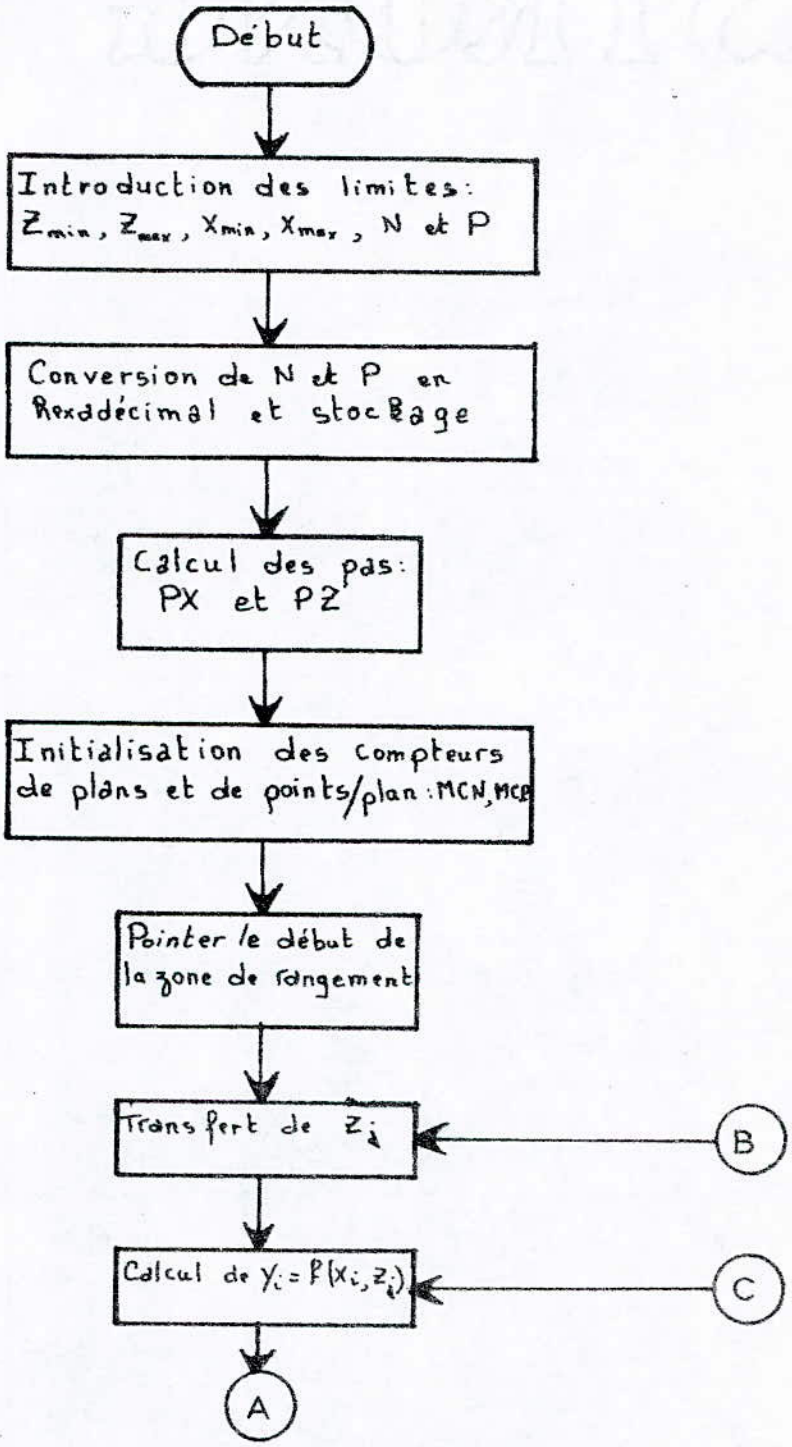
Insérer le Code de LOX

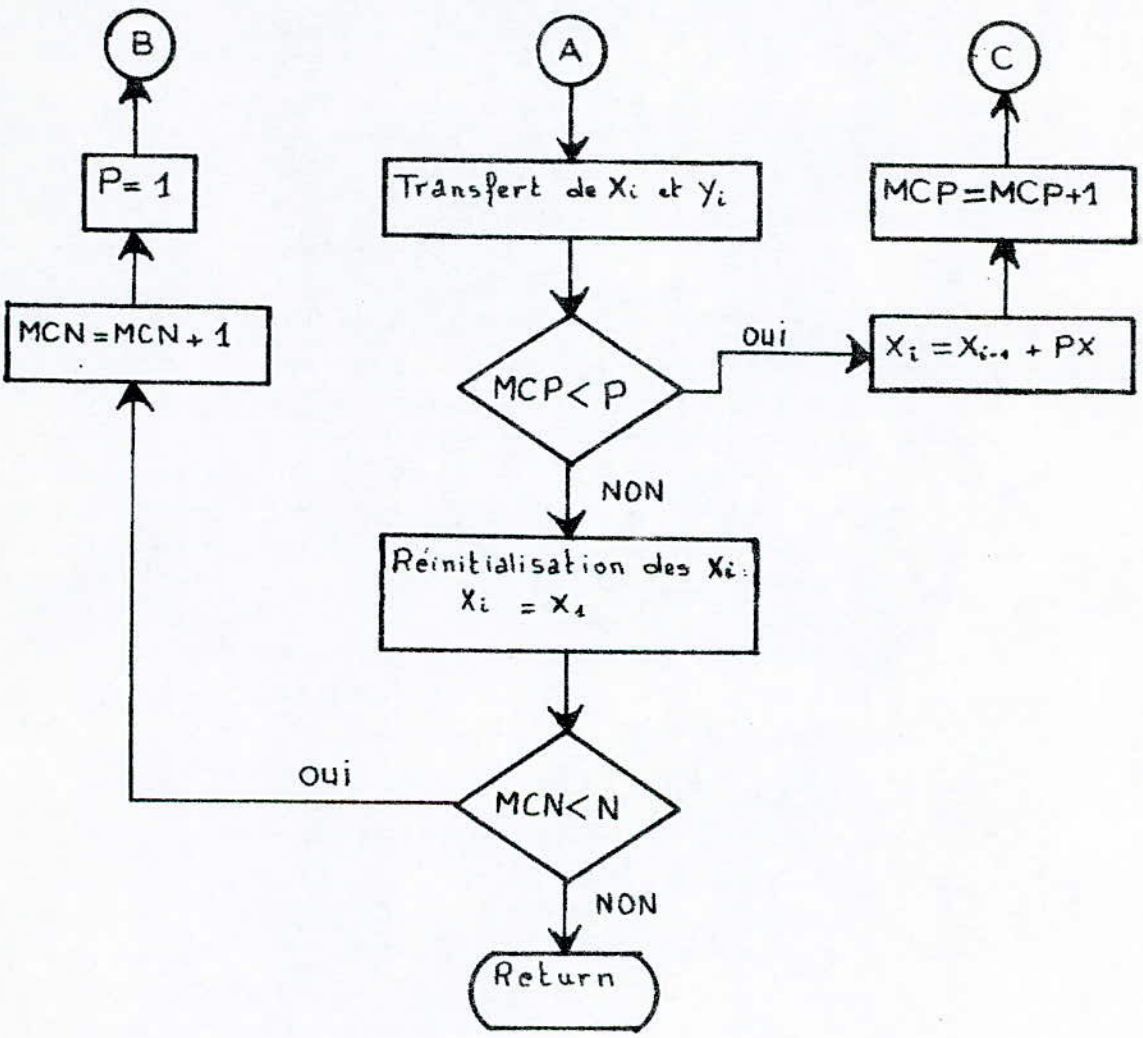

Début
Initialisation des :
- Lignes de crêtes
- paramètres et constantes utilisés
Est-ce une fonction ?
NON
oui
Introduction et stockage des données des tableaux
Calcul et stockage des données de la fonction
Mise à l'échelle
Est-ce une fonction ?
NON
oui
Recherche d'extrema des Z_i : $1 \leq i \leq N$
Recherche d'extrema des X_i : $1 \leq i \leq P$
Recherche d'extrema des Y_i
A



TRACÉ DE LA SURFACE

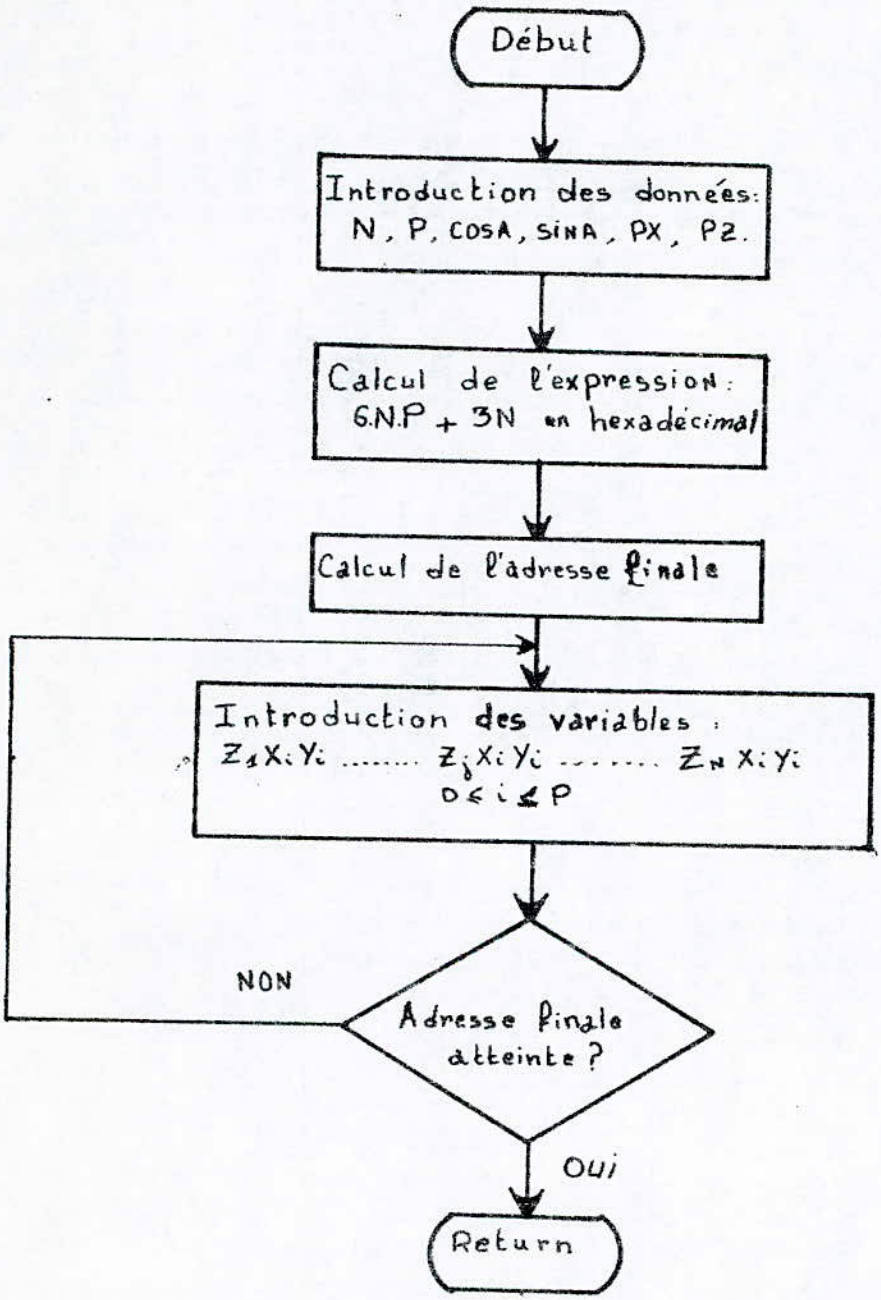
Organigramme.


Début
Introduction des limites: Z_{min}, Z_{max}, X_{min}, X_{max}, N et P
Conversion de N et P en Hexadécimal et stockage
Calcul des pas: PX et PZ
Initialisation des compteurs de plans et de points/plan : MCN, MCP
Pointer le début de la zone de rangement
Transfert de Z_i
B
Calcul de $Y_i = F(X_i, Z_i)$
C
A



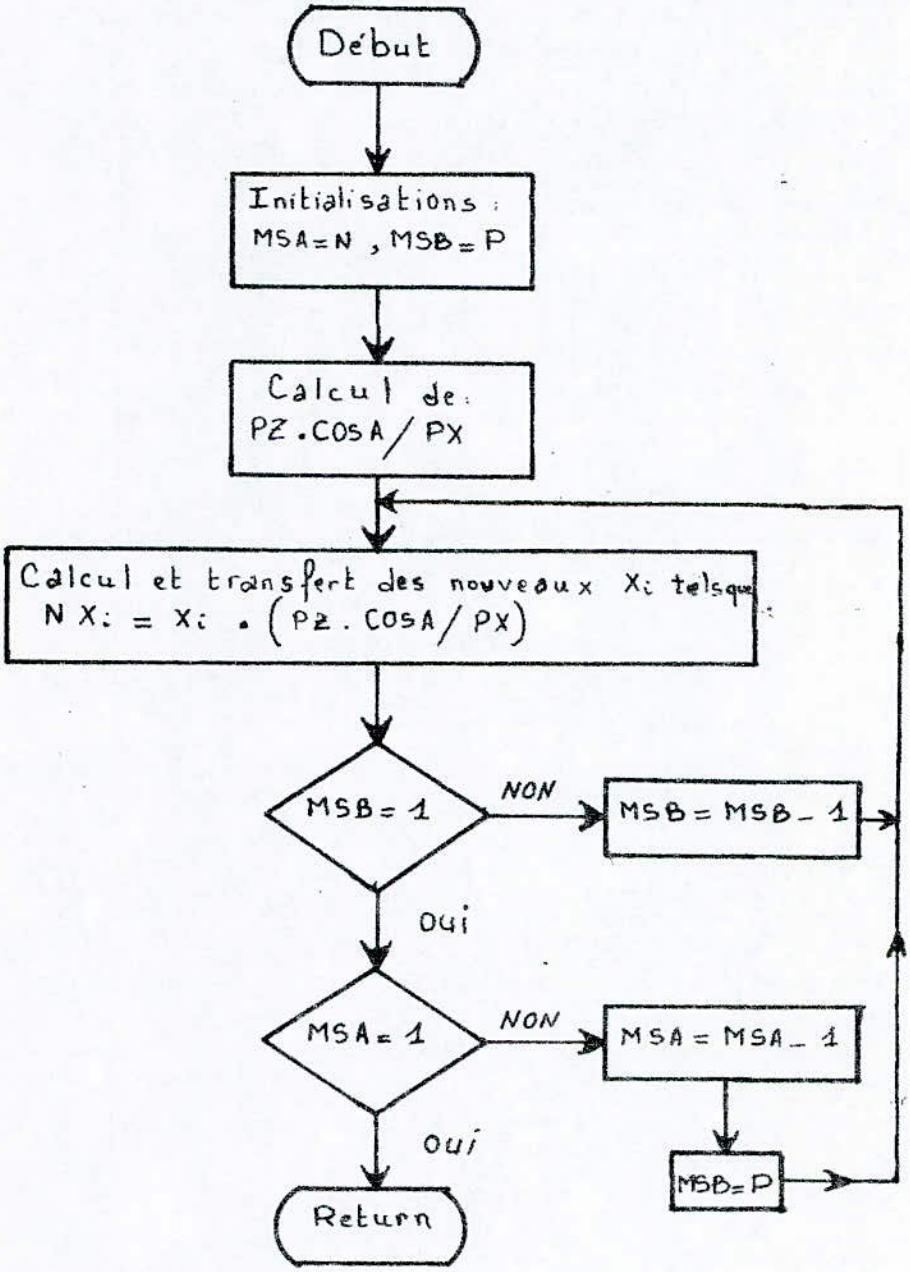
Rangement des données de la fonction

Organigramme



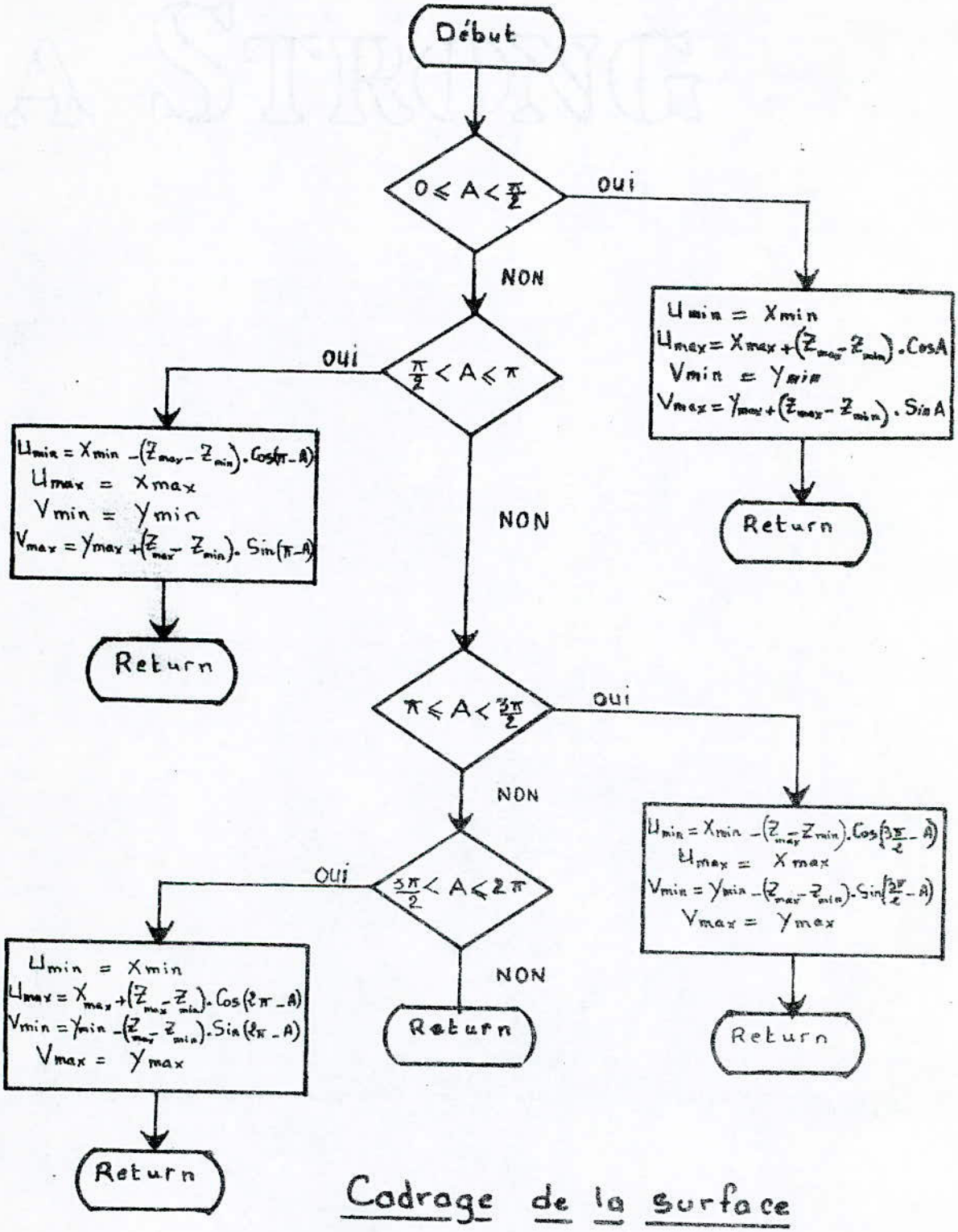
Rangement des données des tableaux

Organigramme



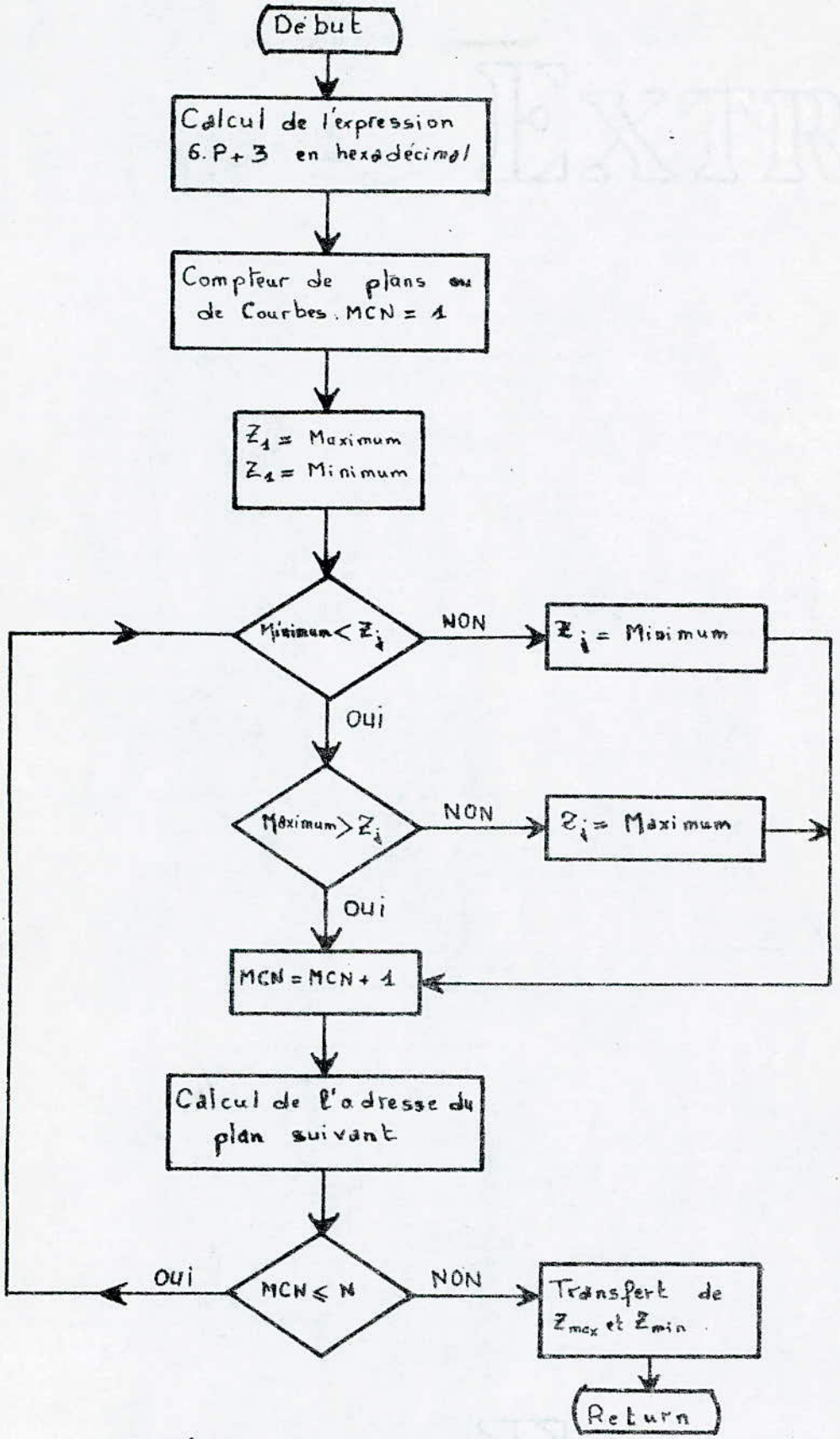
Mise à l'échelle

Organigramme



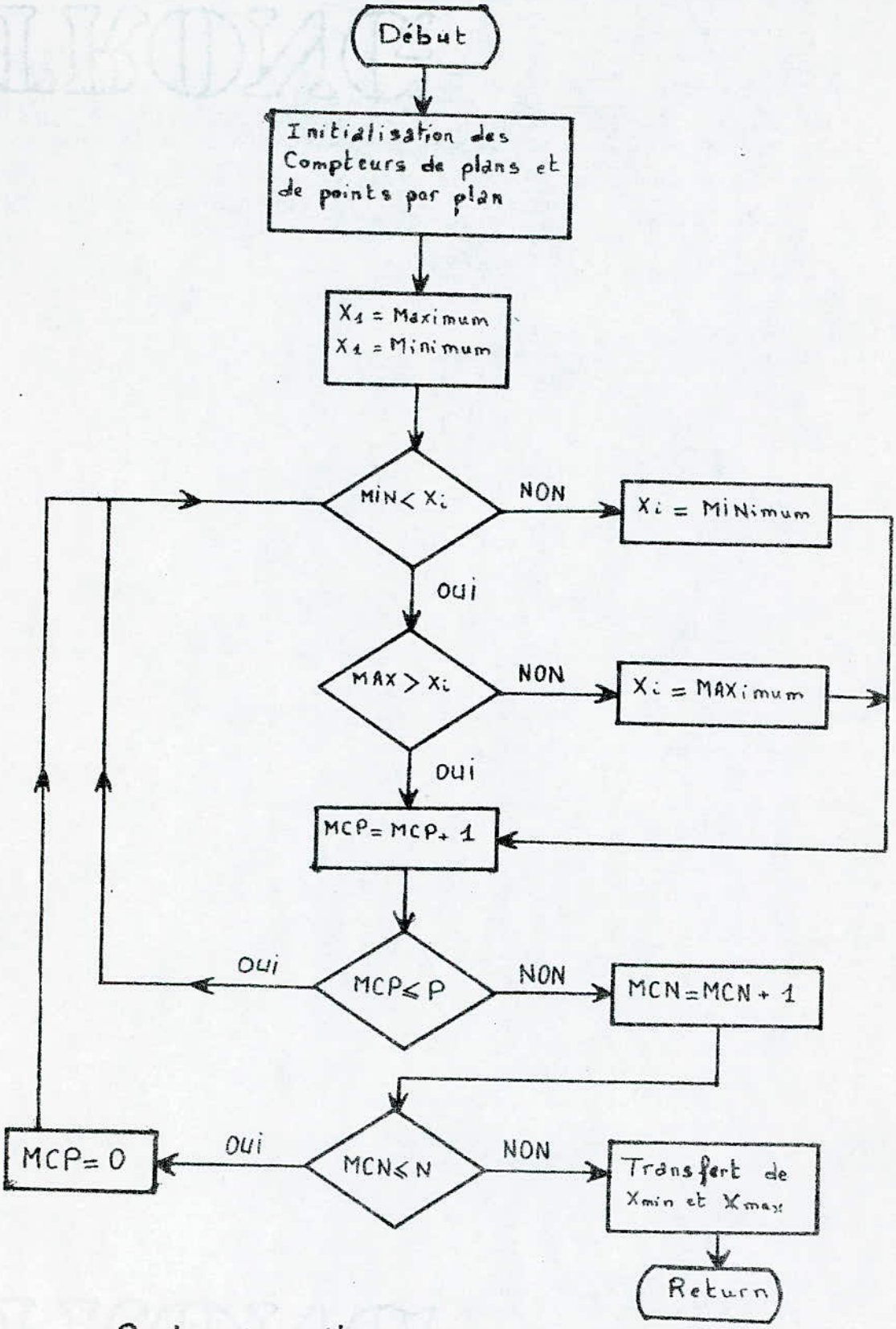
Cadrage de la surface

Organigramme



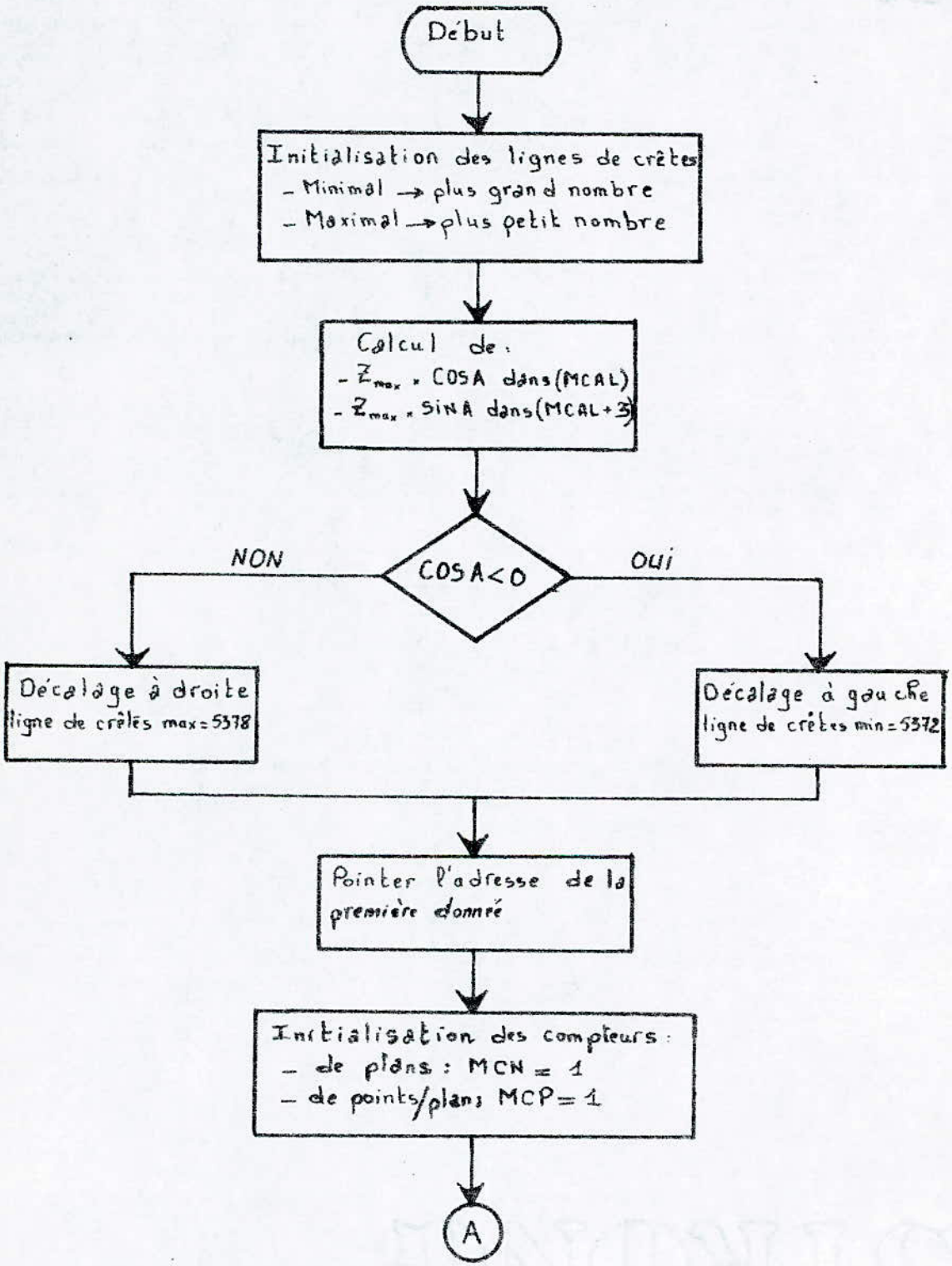
Recherche d'extrèma des Z_j

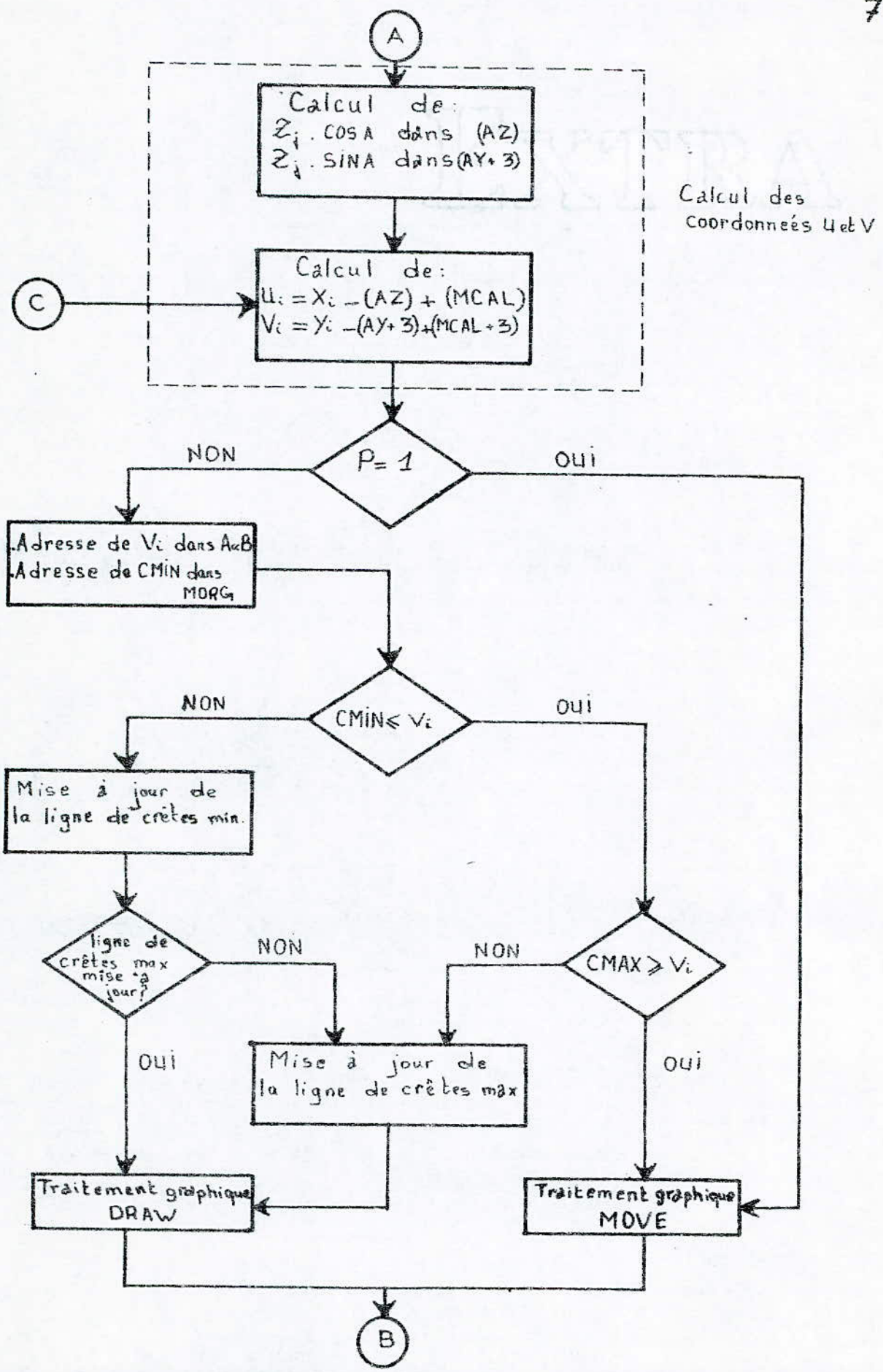
Organigramme

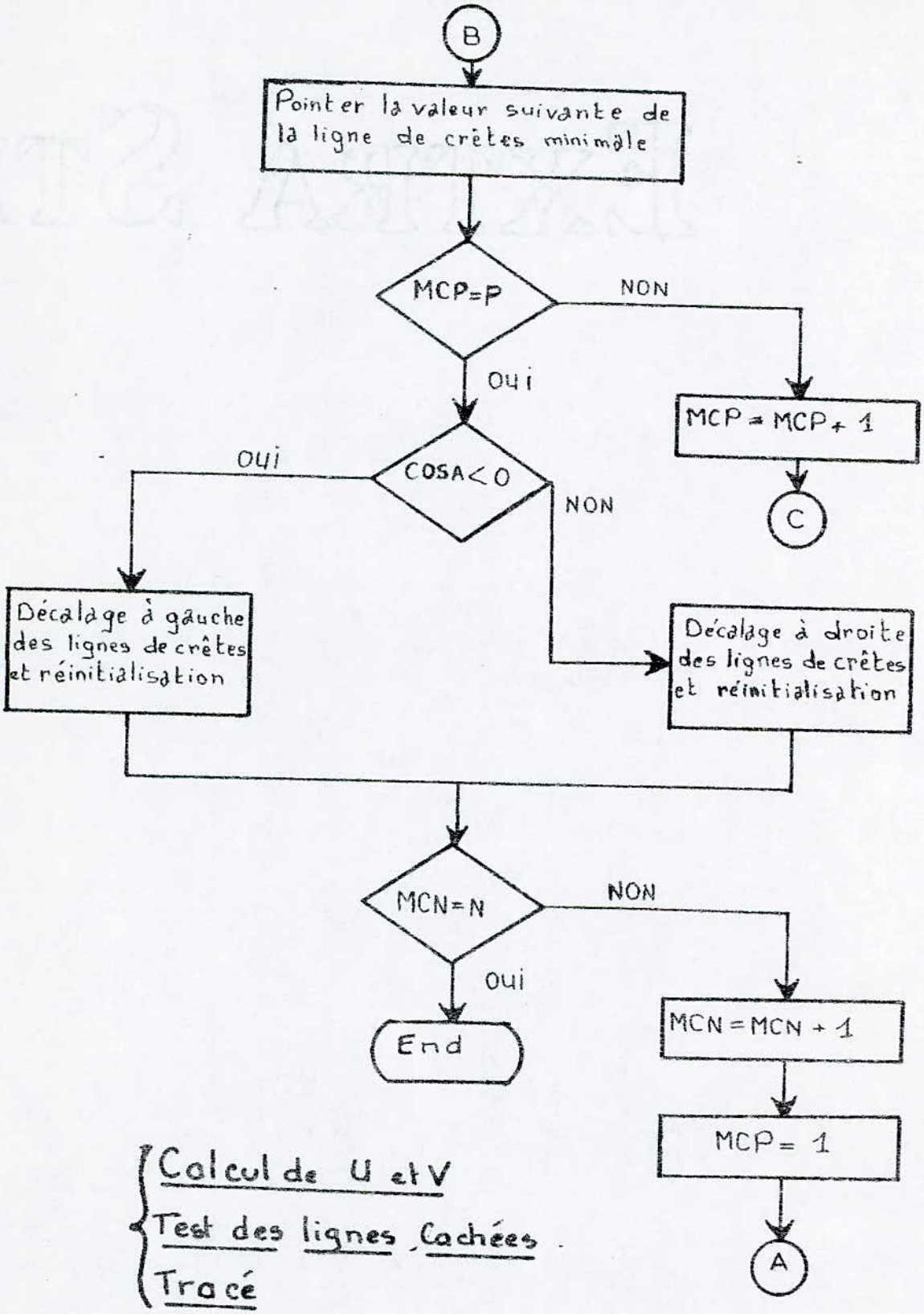


Recherche d'extrèma des X_i

Organigramme


Début
Initialisation des lignes de crêtes
- Minimal → plus grand nombre
- Maximal → plus petit nombre
Calcul de :
- Z_{max} * COSA dans (MCAL)
- Z_{max} * SINA dans (MCAL+3)
COSA<0
NON
OUI
Décalage à droite
ligne de crêtes max=5378
Décalage à gauche
ligne de crêtes min=5372
Pointer l'adresse de la
première donnée
Initialisation des compteurs :
- de plans : MCN = 1
- de points/plans MCP = 1
A


A
Calcul de :
Z_i . COS A dans (AZ)
Z_i . SINA dans (AY+3)
Calcul des coordonnées U et V
C
Calcul de :
$U_i = X_i - (AZ) + (MCAL)$
$V_i = Y_i - (AY+3) + (MCAL+3)$
P = 1
NON
OUI
Adresse de V_i dans A&B
Adresse de CMIN dans MORG
CMIN ≤ V_i
NON
OUI
Mise à jour de la ligne de crêtes min.
ligne de crêtes max mise à jour ?
NON
NON
CMAX ≥ V_i
OUI
Mise à jour de la ligne de crêtes max
OUI
Traitement graphique DRAW
Traitement graphique MOVE
B



- Calcul de U et V
- Test des lignes Cachées
- Tracé

Organigramme

CHAPITRE 5
PERFORMANCES ET RESULTATS

1 Le matériel

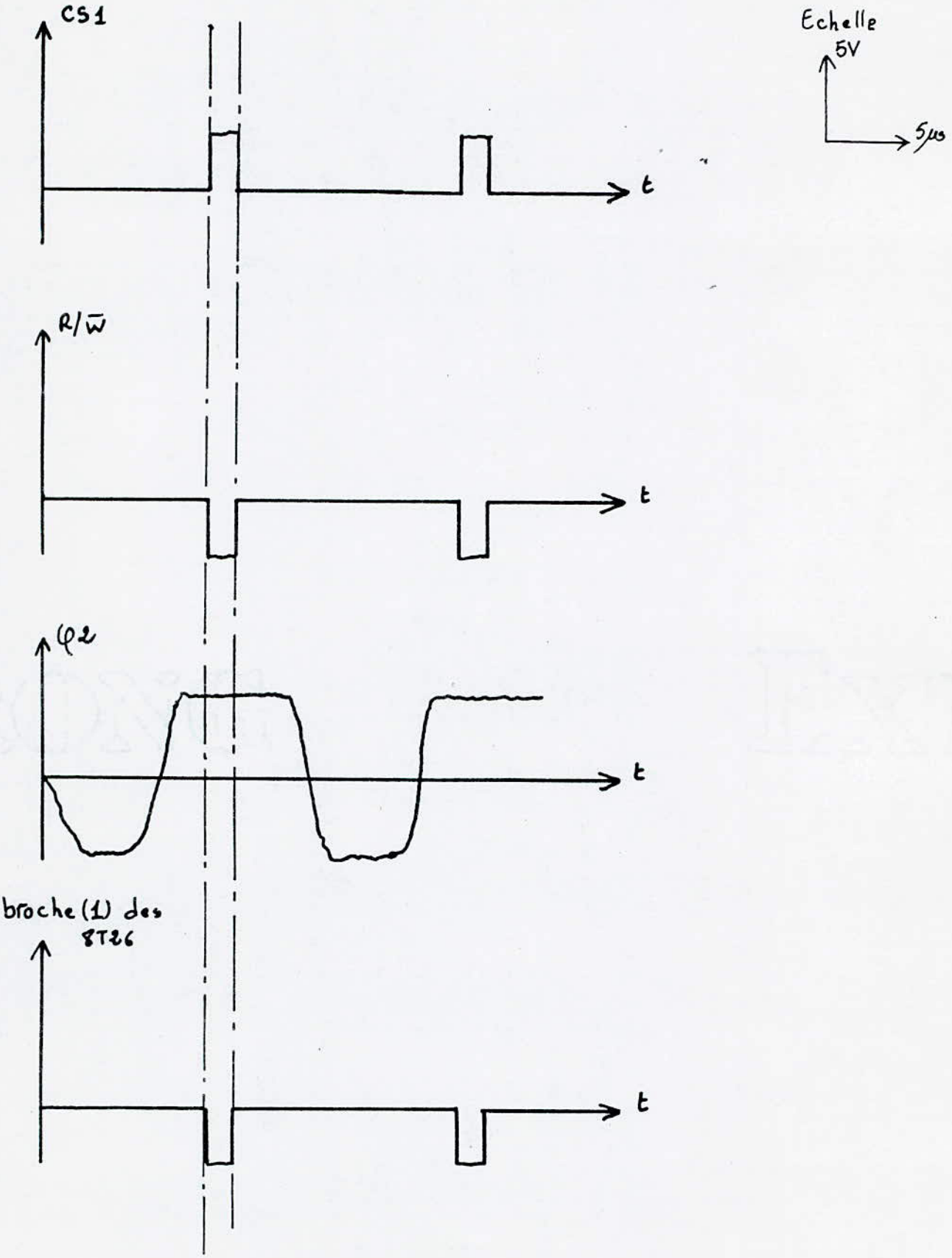
Aprés le cablage de la carte, nous avons procédé à une suite de tests pour s'assurer du bon fonctionnement de celle-ci.

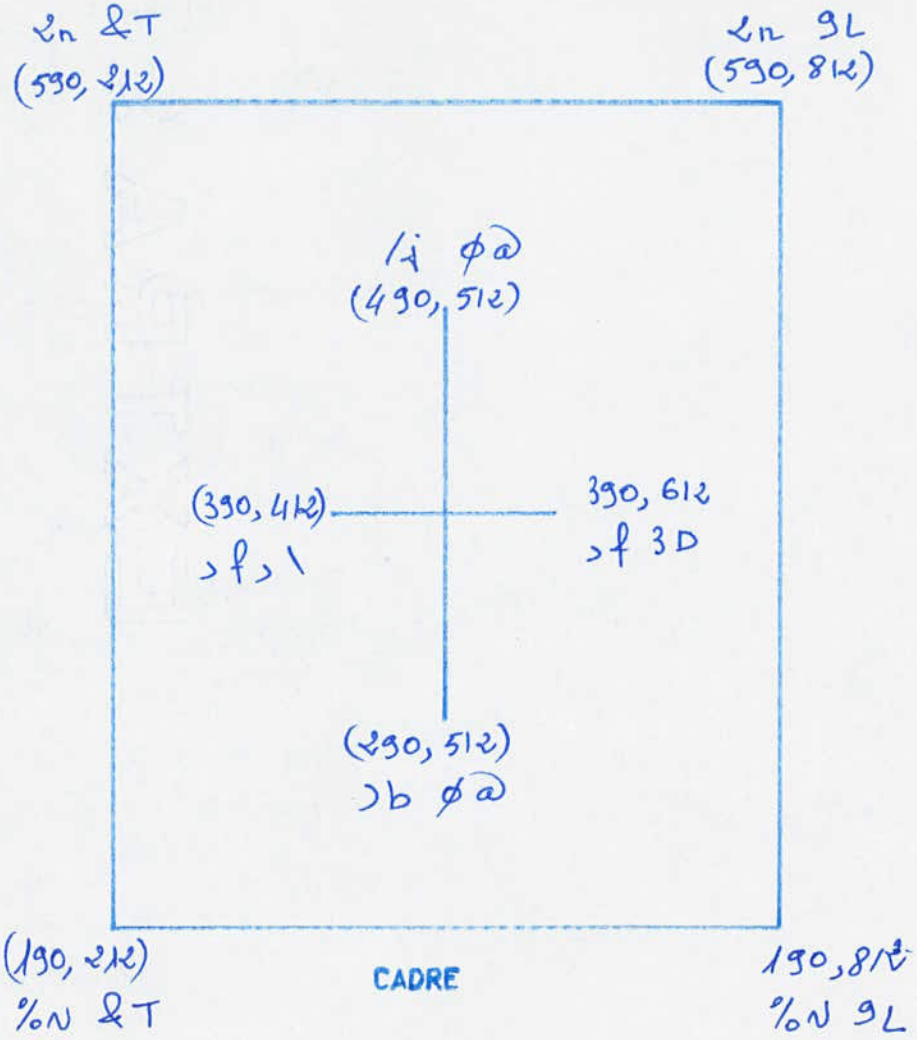
Pour vérifier le signal CS1 de l'ACIA, on envoie à la carte le programme suivant:

```
STA A    $D7F0
BRA
```

De même, pour tester les "chip select" des deux EPROM le même programme est envoyé mais le contenu de l'accumulateur A est alors stocké à l'adresse de début de l'une des EPROM.

Les signaux obtenus sont représentés par la figure suivante.


CS1
t
Echelle
5V
5µs
R/W
t
φ2
t
broche (1) des
8T26
t


&n &T
(590, 212)
&n 9L
(590, 812)
&i ¢@
(490, 512)
(390, 412)
>f,1
390, 612
>f 3D
(290, 512)
>b ¢@
(190, 212)
%N &T
CADRE
190, 812
%N 9L

- fig. 19 -

D'autre part, la réalisation de la carte ayant précédé le développement du logiciel en assembleur, nous avons illustré son bon fonctionnement par le tracé d'un cadre rectangulaire (fig. 19) au moyen du programme suivant:

```
        LDA A   # 03        } Master reset de l'ACIA
        STA A   D7F0        }
        LDA A   # 11        } Configuration de l'ACIA
        STA A   $ D7F0      }
        LDA B   $ 7210
        LDX     # 7211
LP      LDA A   0, X
        JSR     SP
        INX
        DEC B
        BNE     LP
        SWI

SP      STA B   $ 7300      Sauvegarder le contenu de Acc B
LP1     LDA B   $ D7F0
        AND B   # 02        ne laisse passer que le bit b1
        BEQ     LP1
        STA A   D7F1        Donnée à émettre dans TDR.
        LDA B   $ 7300
        RTS
```

Dans ce sous-programme d'émission d'une chaine de caractères :

- les registres d'état et de contrôle sont en $D7F0
- les registres d'émission et de réception en $ D7F1
- la longueur de la liste de caractères en $ 7210
- Enfin, le premier caractère est introduit à l'adresse $7211 en ASCII.

2 - Le logiciel.

L'algorithme que nous avons développé a été mis en œuvre en deux langages : BASIC et ASSEMBLEUR(6800).
Nous nous proposons dans ce chapitre de présenter les résultats obtenus et de les commenter.

2-1 Le programme écrit en basic permet de visualiser des surfaces qui représentent aussi bien des fonctions de deux variables $y = f(x, z)$ que des tableaux de mesures expérimentales.

. Plusieurs vues en perspective peuvent être tracées sur une même feuille. Chacune d'elles peut représenter la surface :

* Vue sous un certain angle.
* translatée dans l'espace.

* agrandie ou réduite suivant que l'on veuille mieux percevoir un détail ou avoir une vue globale.

. le tracé peut être complété (au gré de l'opérateur) par la représentation des repères U et V (dans le plan) et X, Y, Z (dans l'espace).

. La surface peut être représentée par 31 plans de 37 points chacun au maximum (à l'aide du TEKTRONIX 4051)

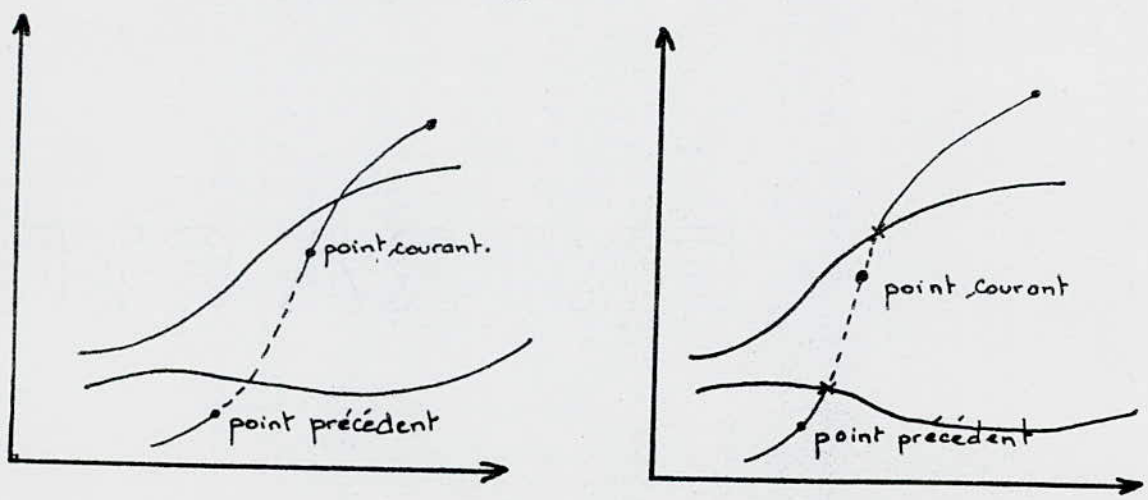
Mais, malgré la souplesse d'utilisation qui le caractérise et bien que les résultats obtenus soient satisfaisants, l'exécution de ce programme intéractif nécessite un temps important (environ 45 mn pour tracer une surface définie par 31 plans de 37 points).

2-2 <u>Le programme écrit en assembleur</u> ne permet de représenter que les surfaces générées par des tableaux de valeurs expérimentales (Toutefois, pour mettre au point le programme, nous avons prévu de tracer la surface représentant la fonction : $y = (4-z)^2/2 + 4 - x^2/10$).

- chaque surface peut être tracée avec, au maximum, 35 courbes de 100 points chacune.
- Enfin, le programme se caractérise par son extrême rapidité d'execution.

Cependant, il est important de remarquer que

* Quel que soit le langage de programmation, le test des lignes cachées est d'autant moins rigoureux que la résolution maximale n'est pas utilisée. (fig 20)
* En raison de l'utilisation d'une perspective cavalière pour effectuer le passage de l'espace à trois dimensions vers le plan et de la mise à l'échelle, une surface ne peut être représentée avec un angle de visée $A = 90°$ (le pas sur X serait alors nul (relation 2-4))



Ce qu'on a.

Ce qu'on devrait avoir

- figure 20 -

* Nous avons envisagé d'effectuer une rotation de la surface. Deux problèmes essentiels se sont posés :

1. Lorsqu'on effectue une rotation de la surface autour de de l'axe X ou Y, les points qui appartenaient à un même plan z = constante, appartiennent alors à des plans différents. D'où la nécessité de rechercher les points ayant la même valeur de z puis de les ranger suivant les x croissants et les z décroissants.

2. Considérons la surface représentée ci dessous (fig. a) Si on lui fait subir une rotation autour de l'axe Z par exemple, la surface obtenue est celle indiquée à la figure b. Le test des lignes cachées est alors erroné car il ne peut s'appliquer à des fonctions non injectives.

fig. a

Test correct.
La fonction étant injective, il suffit de comparer les ordonnées des points

fig. b

Test erroné.
La fonction n'est pas injective. Il faut donc stocker aussi bien les abscisses que les ordonnées puis d'effectuer la comparaison.

Liste des surfaces représentées

figure 21 La fonction représentée est :

$$y = 40 \sin(0{,}043 \sqrt{x^2 + z^2})$$

La surface a été tracée avec une perspective de 60° et en ne manipulant que les valeurs entières.

figure 22 La même fonction est représentée avec une perspective de 120°. Cependant, si les courbes tracées sont visiblement plus lisses, c'est que des valeurs réelles de U et V sont utilisées.

figure 23 Plusieurs vues en perspective peuvent être affichées (quatre dans ce cas)

figure 24 On peut visualiser une surface pour se faire une idée de son apparence finale. Mais une fois tracée, on peut vouloir la modifier....
(C'est toujours la même surface qui est représentée. Seules les limites de X et de Z changent.)

figure 25

Une autre fonction est représentée

Il s'agit de $y = (4-z)^2/2 + 4 - x^2/10$

figure 26

La fonction représentée est :

$$y = 1,5 * (x^2+y^2) \sqrt{1-(x^2+y^2)}$$

avec une perspective de 45°.

figure 27

La même fonction est représentée.

En changeant de perspective, d'autres détails apparaissent.

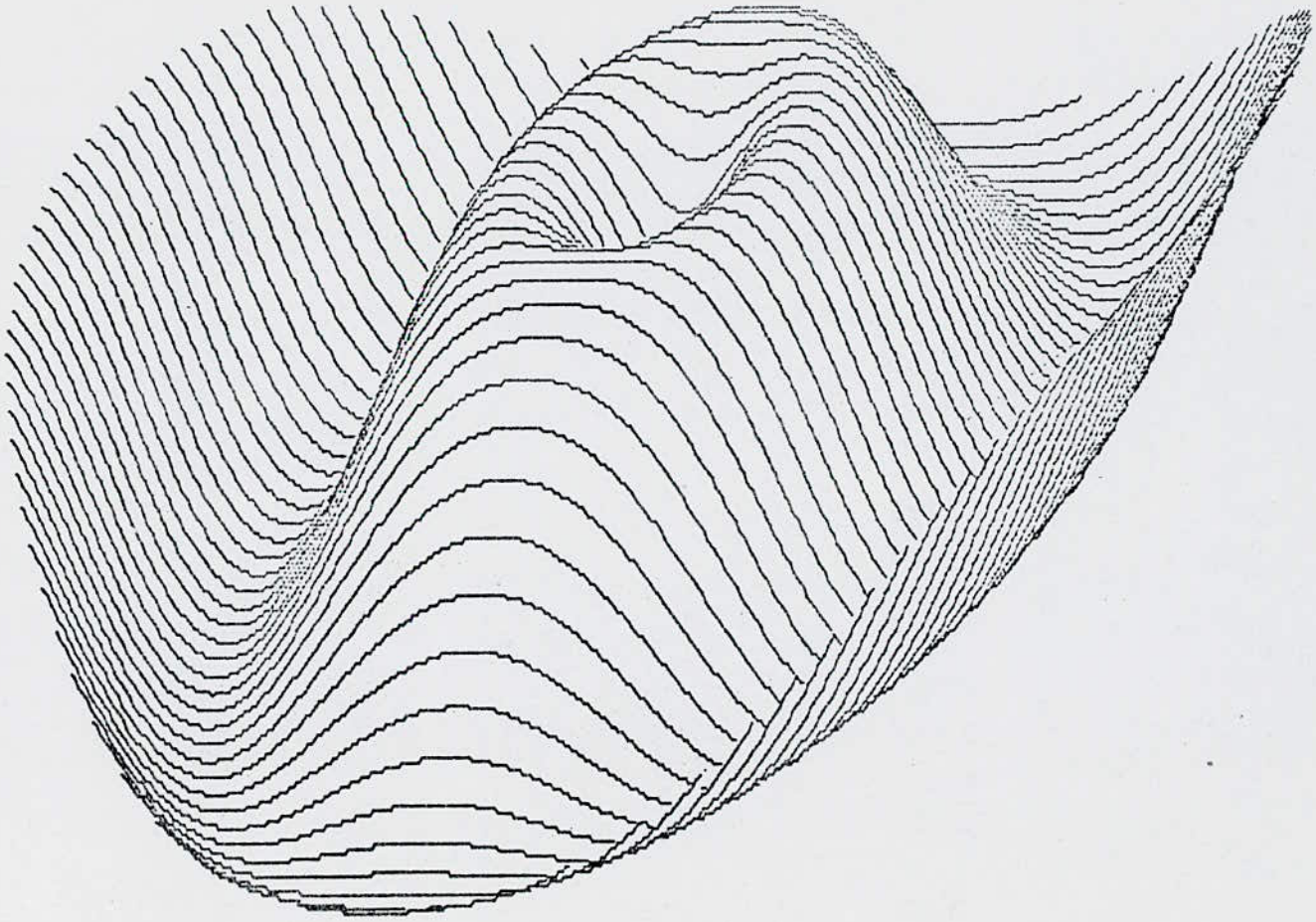
figure 28

La surface visualisée, dans cette figure, représente la fonction : $y = 8 * (\sin \sqrt{x^2+z^2}) / \sqrt{x^2+z^2}$

figure 29

On représente, dans cette figure, avec une perspective à 45°, la fonction :

$$y = 100 * \text{SIN}((X-800)/800) * \text{COS}((z-800)/800 + 1.5) + 410$$



A = 60°

Fig. 21

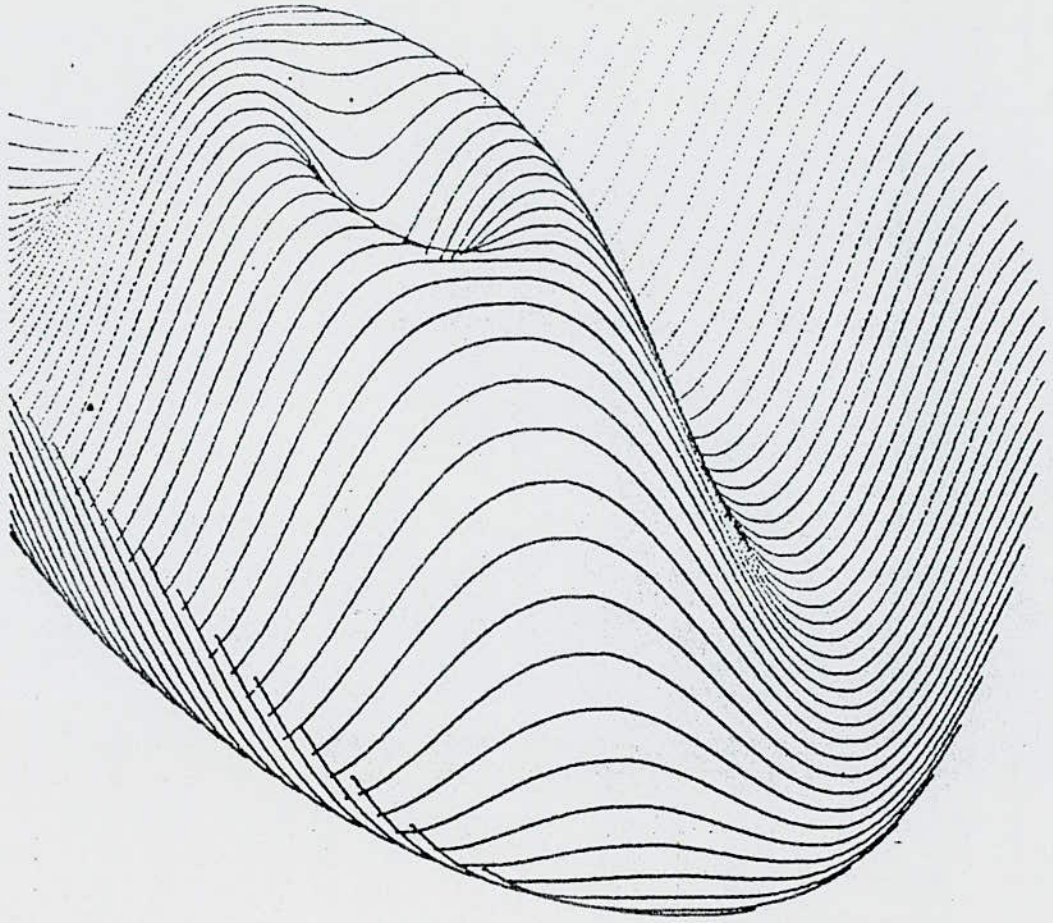
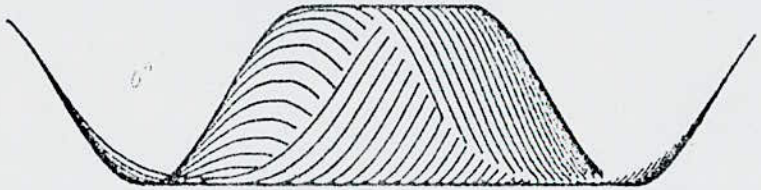
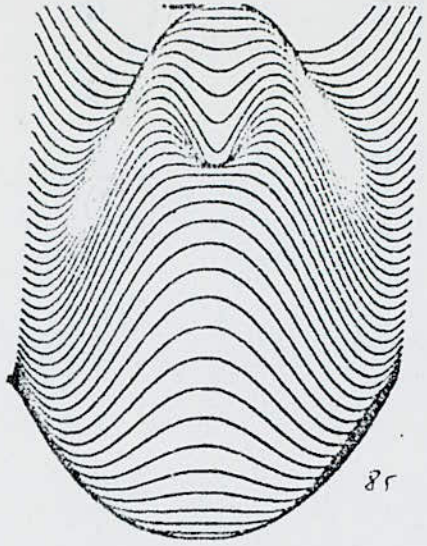


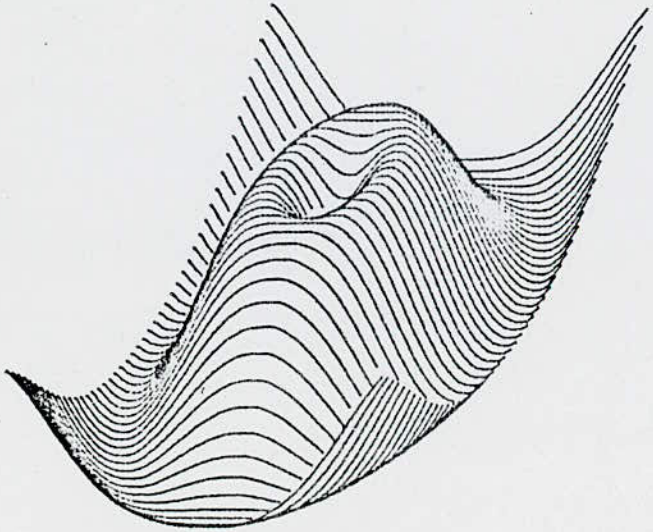
Fig. 22



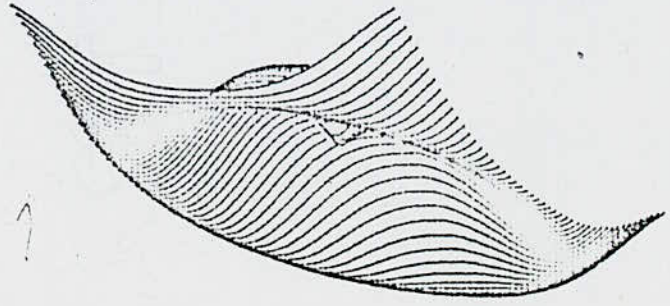
A = 0°



A = 85°



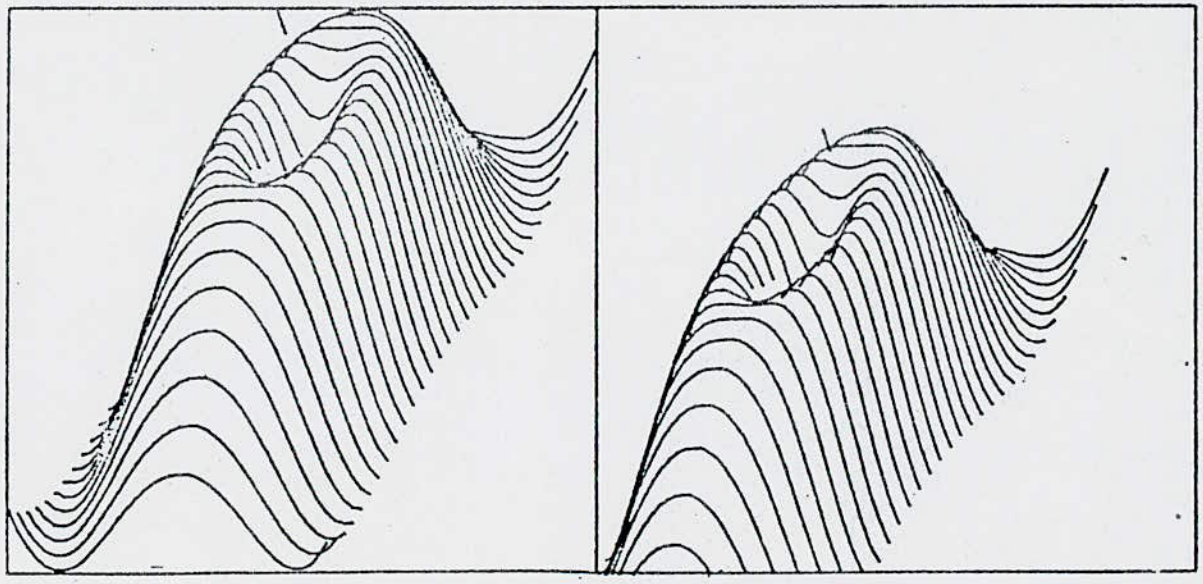
A = 45°



A = 315°

Fig. 23

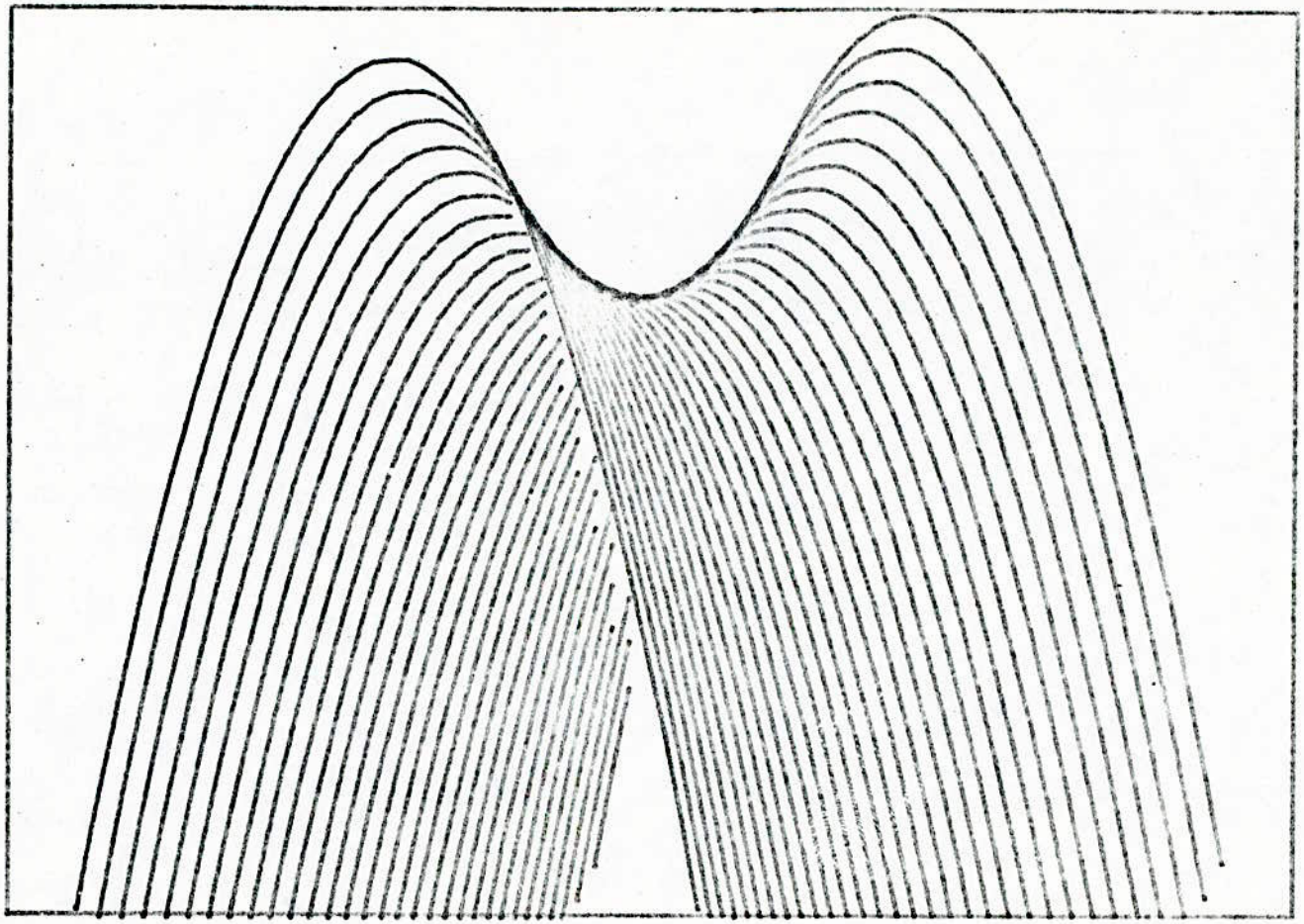
A = 45°



Vue en perspective

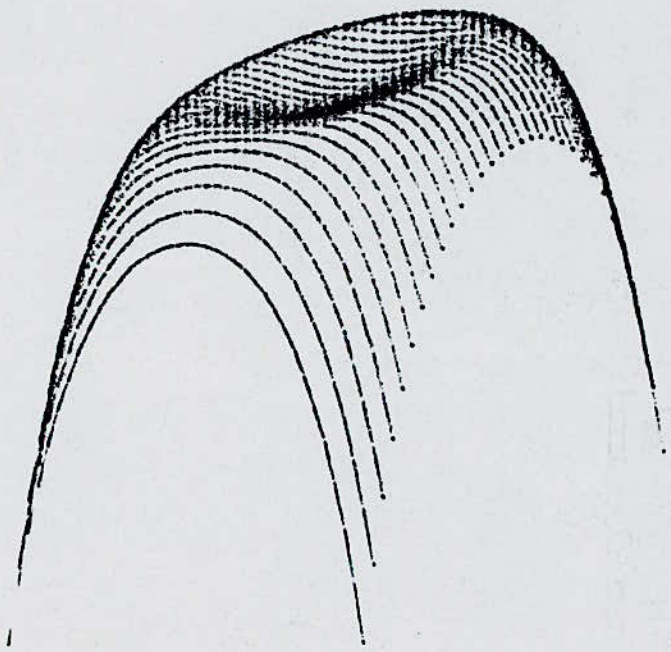
translation de vecteur
(50, 50, 50)

Fig24



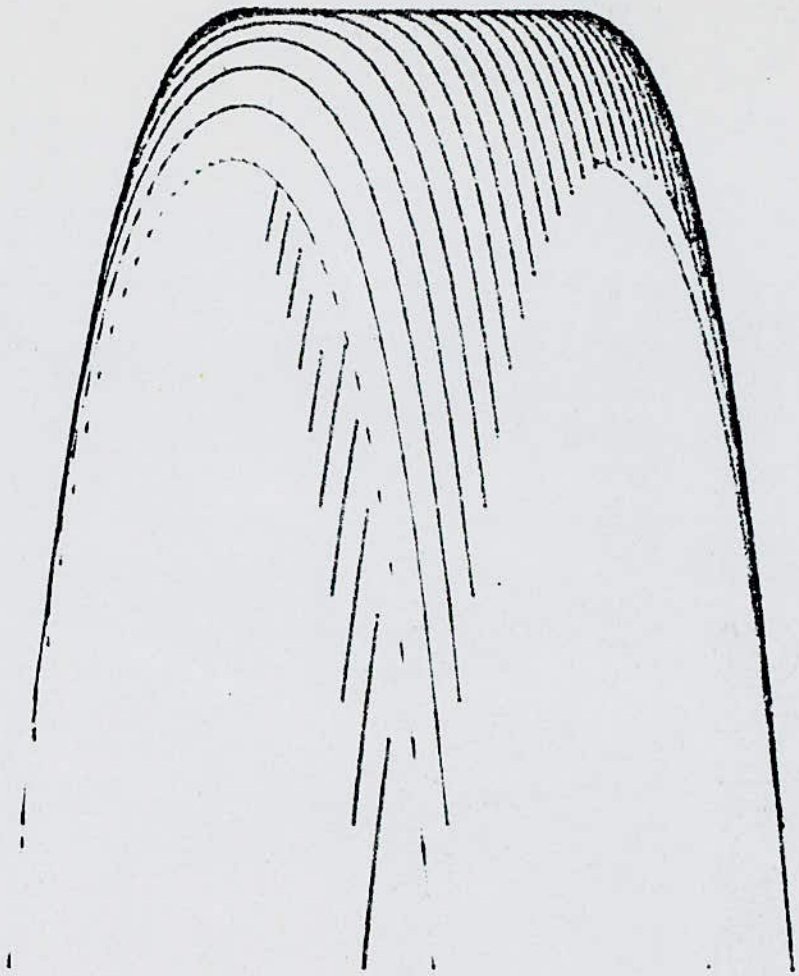
A=45

- fig. 25 -



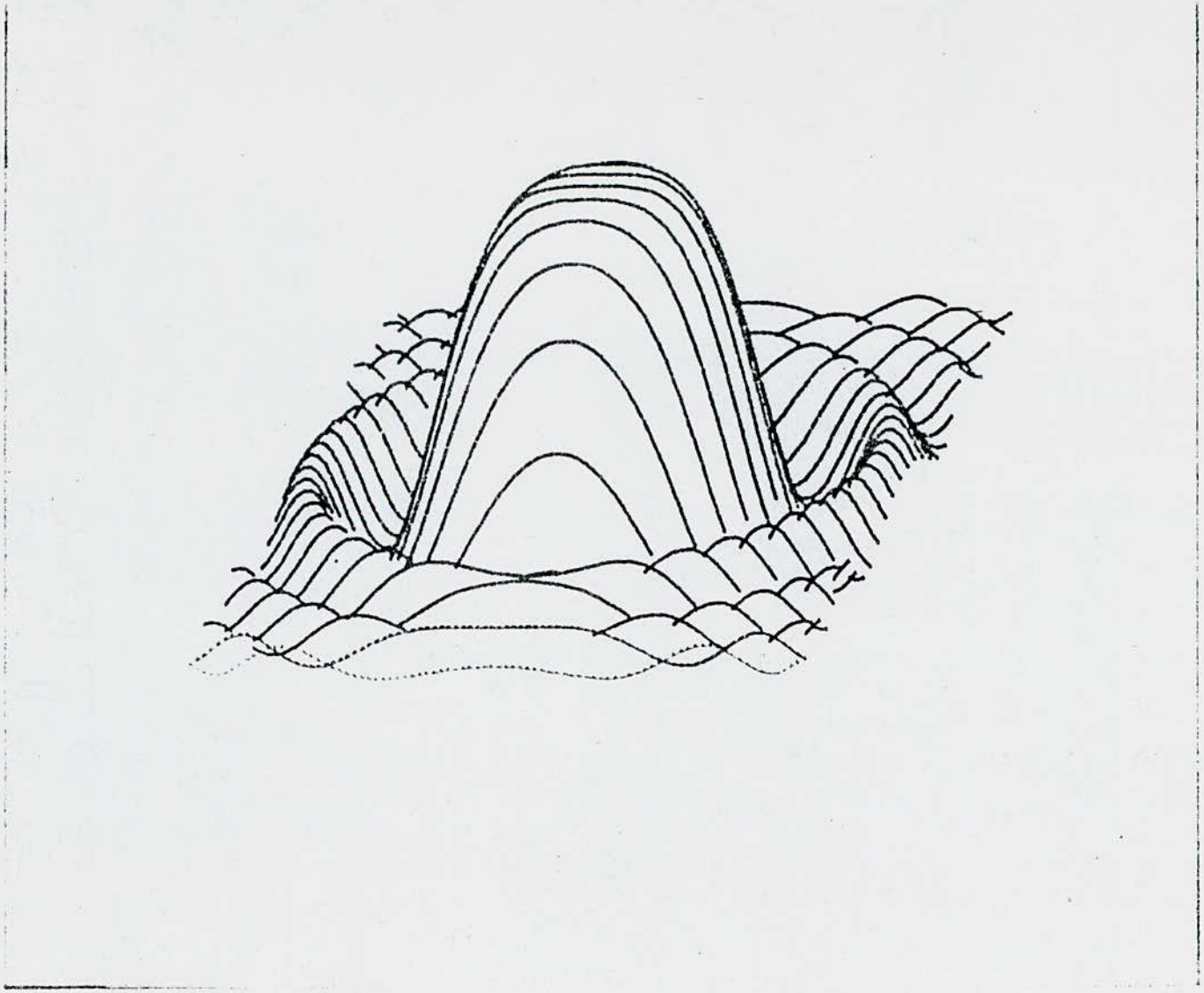
A=45

Fig. 26



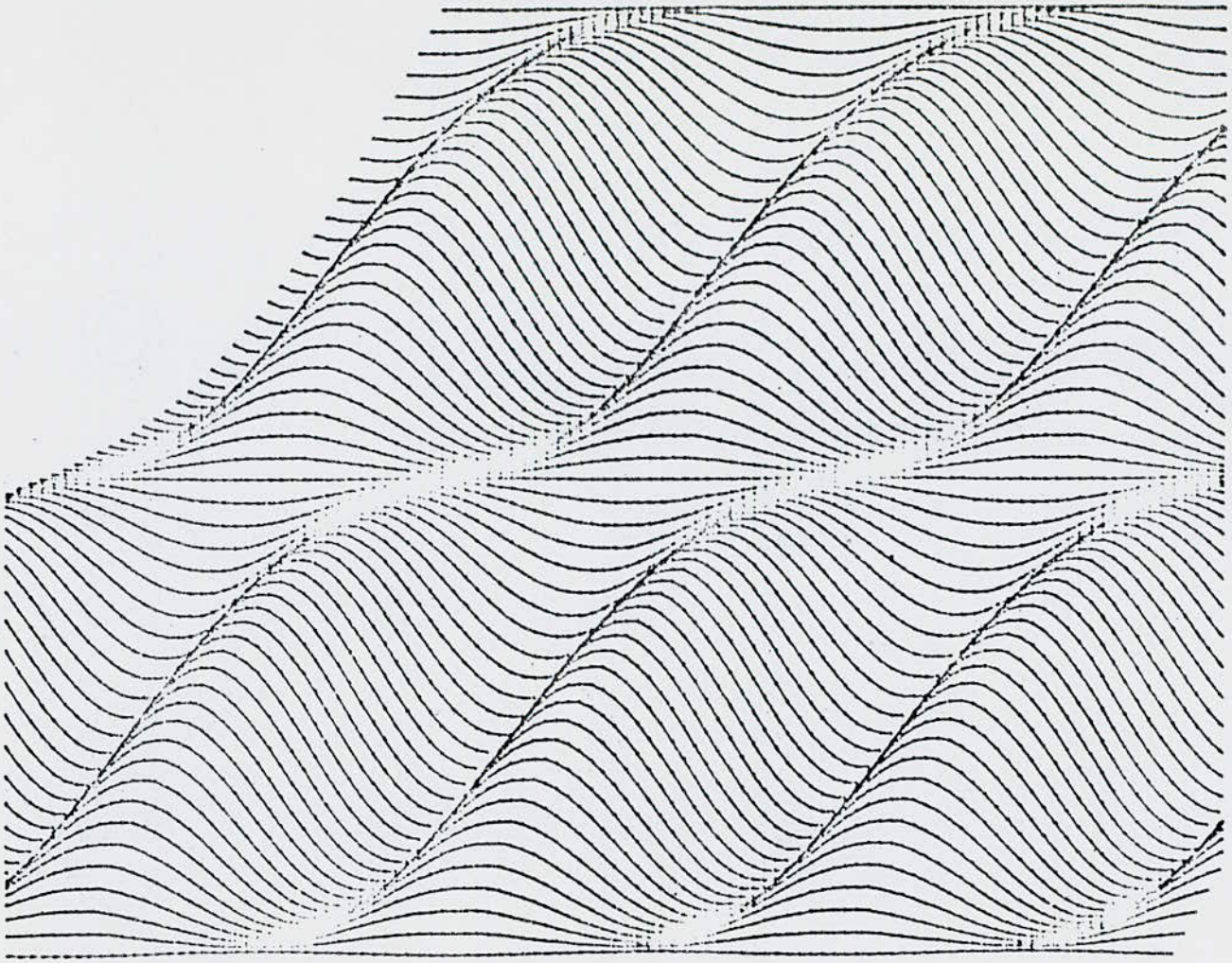
A = 0

Fig. 27



A=45

Fig. 28



A = 45°

Fig. 29

Conclusion

Cette étude présente une solution suffisante pour la représentation tridimentionnelle des fonctions de deux variables avec l'elimination des parties cachées. Cependant, quelques améliorations logicielles peuvent être apportées.
En effet, il serait intéressant de :

- relever le problème de la limitation du choix de la perspective posé par la mise à l'échelle effectuée lors du tracé. Ce problème étant lié à la représentation de l'espace à trois dimensions dans le plan, il serait possible d'envisager un traitement particulier de l'angle $A = 90°$ ou d'utiliser une perspective vraie ce qui aurait pour effet d'augmenter le réalisme de l'image.

- Améliorer la qualité du tracé. Les défauts des courbes étant dûs au principe du test des lignes cachées, nous proposons d'établir un algorithme de recherche des points d'intersection avec la portion de surface déja tracée et le compléter par des calculs d'interpolation.

ANNEXE

1_ Etude de l'ACIA

2_ La norme RS232

3_ Caracteristiques de la table traçante TEKTRONIX 4662

4_ Caractéristiques de la bibliothèque mathématique.

5_ Brochages

1 - Etude de l'ACIA

L'ACIA (asynchronous communication interface adapter) est un circuit réalisé en technologie N-MOS. Toutes ses entrées/sorties sont compatibles T.T.L.

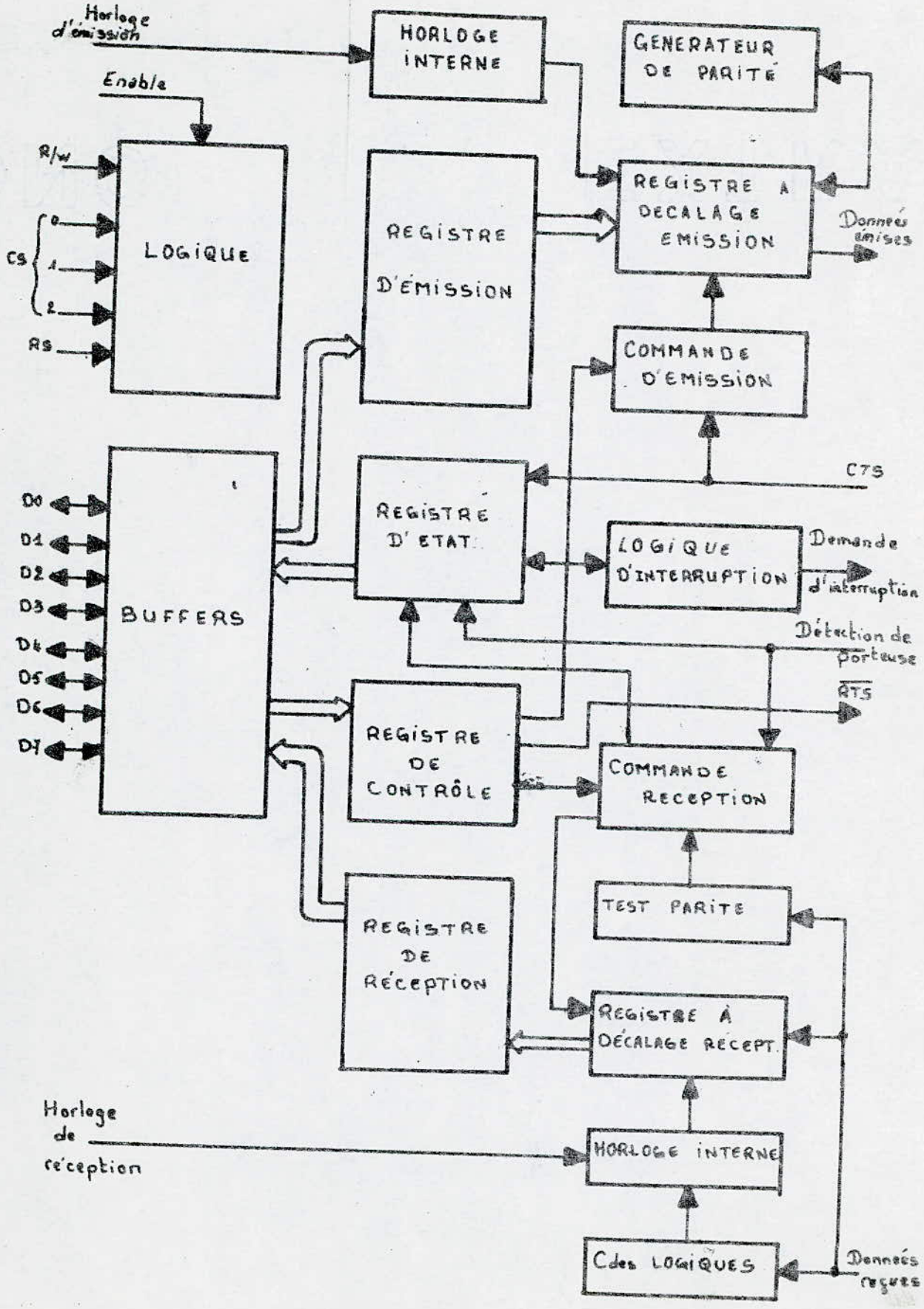
1-1 Synoptique interne de l'ACIA

Ce circuit regroupe dans un boitier à 24 pattes :

- un émetteur de données asynchrones
- un récepteur de données asynchrones
- Une logique de commande.
- Des entrées d'horloge séparées pour l'émetteur et le récepteur
- 4 registres de dialogue et de contrôle.

La pièce maîtresse des sous-ensembles d'émission et de réception est un registre à décalage. Celui-ci reçoit bien sûr, une horloge de l'extérieur après passage dans une logique qui permet d'en diviser la fréquence par 1, 16, ou 64, et un générateur de parité lui est associé.

Les données provenant du bus du microprocesseur sont mémorisées dans le registre de transmission de données (TDR) et sont converties du mode série au mode parallèle. Ensuite, ces données sont disponibles dans le registre de réception de données d'où elles sont envoyées


Horloge d'émission
Enable
R/W
CS
0
1
2
RS
LOGIQUE
HORLOGE INTERNE
GENERATEUR DE PARITÉ
REGISTRE D'ÉMISSION
REGISTRE A DECALAGE EMISSION
Données émises
COMMANDE D'EMISSION
CTS
D0
D1
D2
D3
D4
D5
D6
D7
BUFFERS
REGISTRE D'ETAT
LOGIQUE D'INTERRUPTION
Demande d'interruption
Détection de porteuse
RTS
REGISTRE DE CONTRÔLE
COMMANDE RECEPTION
TEST PARITE
REGISTRE DE RÉCEPTION
REGISTRE À DÉCALAGE RECEPT.
Horloge de réception
HORLOGE INTERNE
Cdes LOGIQUES
Données reçues

- figure 1 -

vers le bus du microprocesseur (fig. 1)

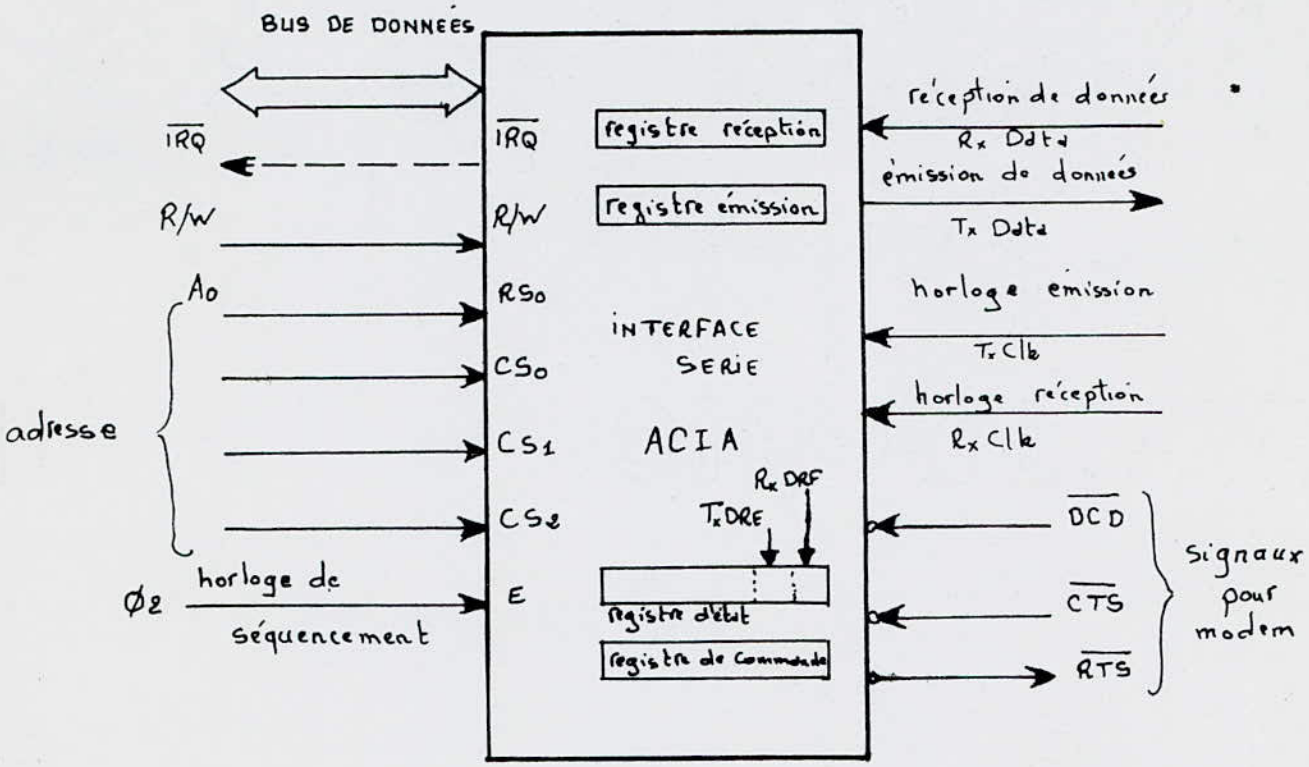
1.2 Adressage des registres

Ainsi que nous l'avons indiqué dans le paragraphe précédent, les registres internes de l'ACIA sont au nombre de quatre ; Ceux-ci ne pouvant être que lus ou écrits, ce qui réduit l'adressage au tableau ci-après :

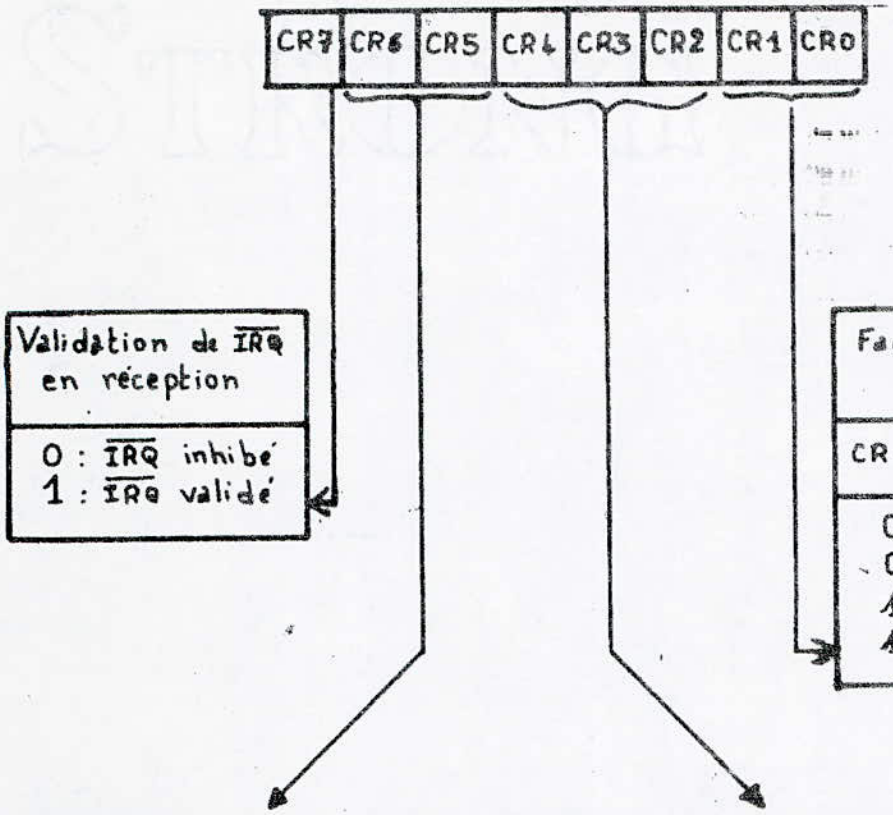
RS	R/W	Registre Selectionné	Accés en
0	0	Registre de Controle	Ecriture
0	1	Registre d'état	Lecture
1	0	Registre d'émission	Ecriture
1	1	Registre de réception	Lecture

1.3 Les signaux de l'ACIA

Ces signaux sont à scinder en deux groupes : les signaux côté microprosseur et les signaux côté exterieur. Nous allons les représenter dans le schéma ci-dessous.

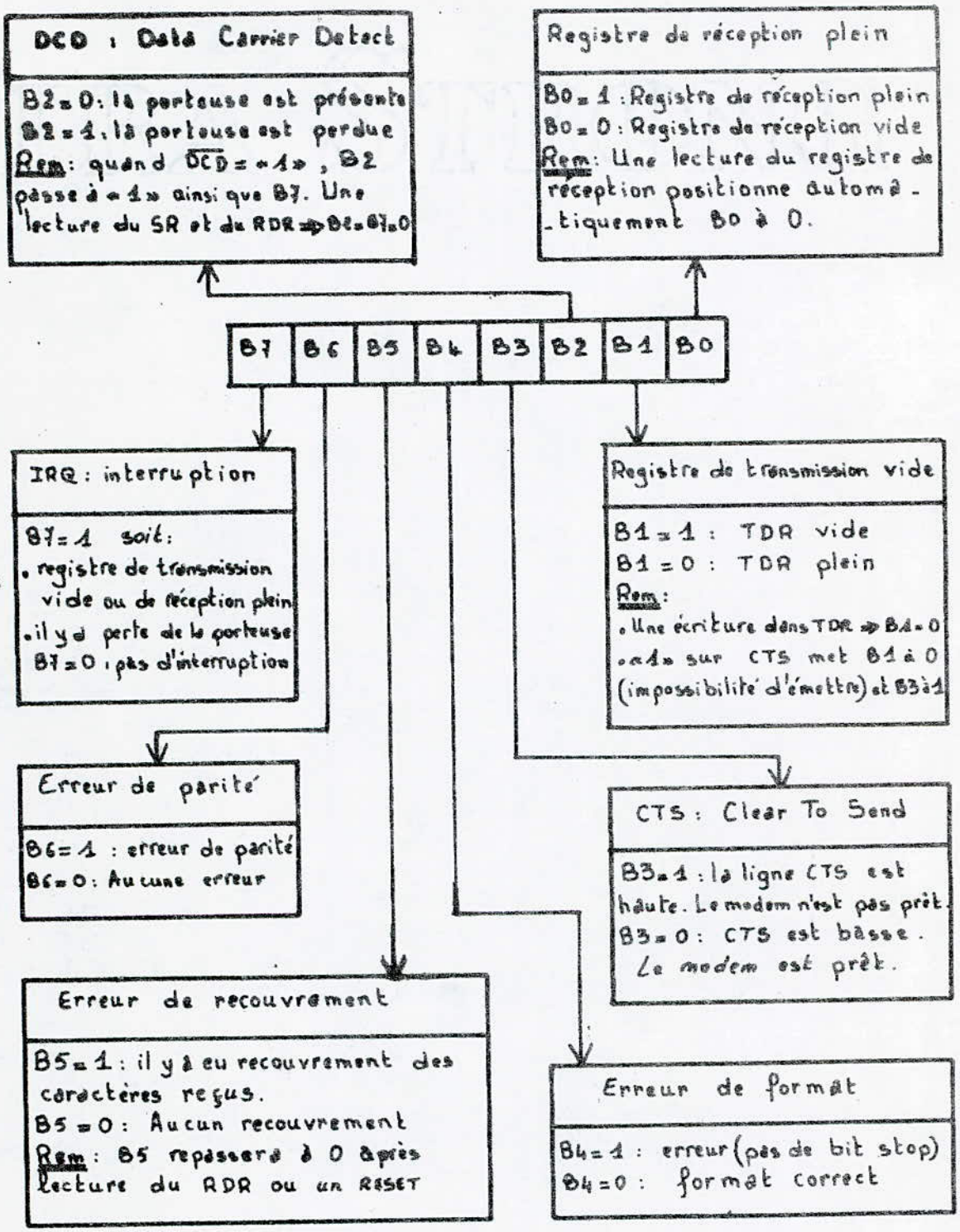


Interface série programmable du 6800 (ACIA)



Registre de Controle

Determination du mot de commande



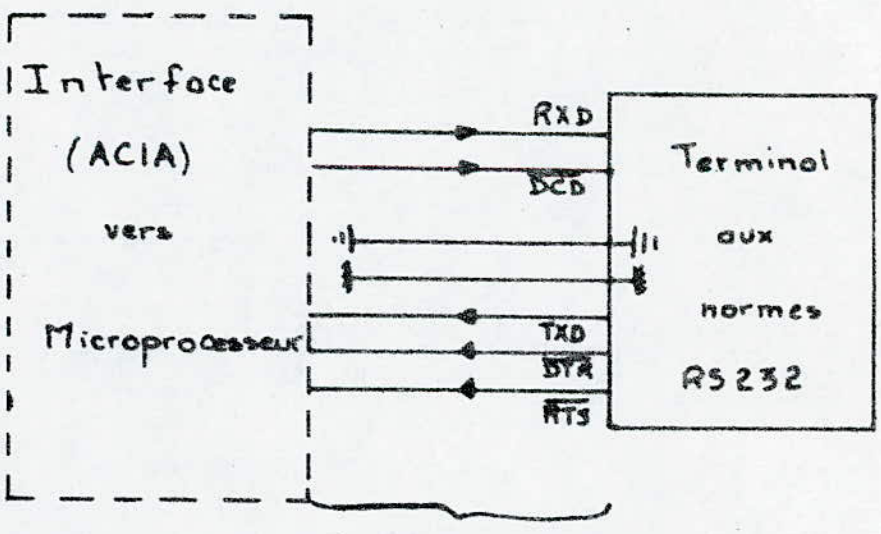
Registre d'état

MOT d'état de l'ACIA

2. La norme RS232

La norme RS 232 est la plus répandue à l'heure actuelle puisque quasiment tous les équipements informatiques disposent d'une liaison série asynchrone à la norme RS 232. Cette dernière définit non seulement des niveaux électriques sur les lignes de transmission série, mais aussi un certain nombre de signaux de contrôle que l'utilisateur est libre d'exploiter ou non.

Ci-dessous, sont représentées les lignes les plus utilisées lors de la connexion d'un terminal à un microcalculateur:



Liaison RS 232

* RXD (Receive Data) :

réception des données sous forme de niveaux de tension où le zéro logique correspond à une tension comprise entre 5V et 15V tandis que le '1' correspond à un niveau compris entre -15V et -5V

* TXD (Transmit Data) :

émission des données sous la même forme que pour une réception

* CTS (clear to send) :

le modem est prêt à émettre.

* RTS (Request to send) :

le microcalculateur signale au modem qu'il est prêt à émettre .

* DTR (Data terminal Ready) :

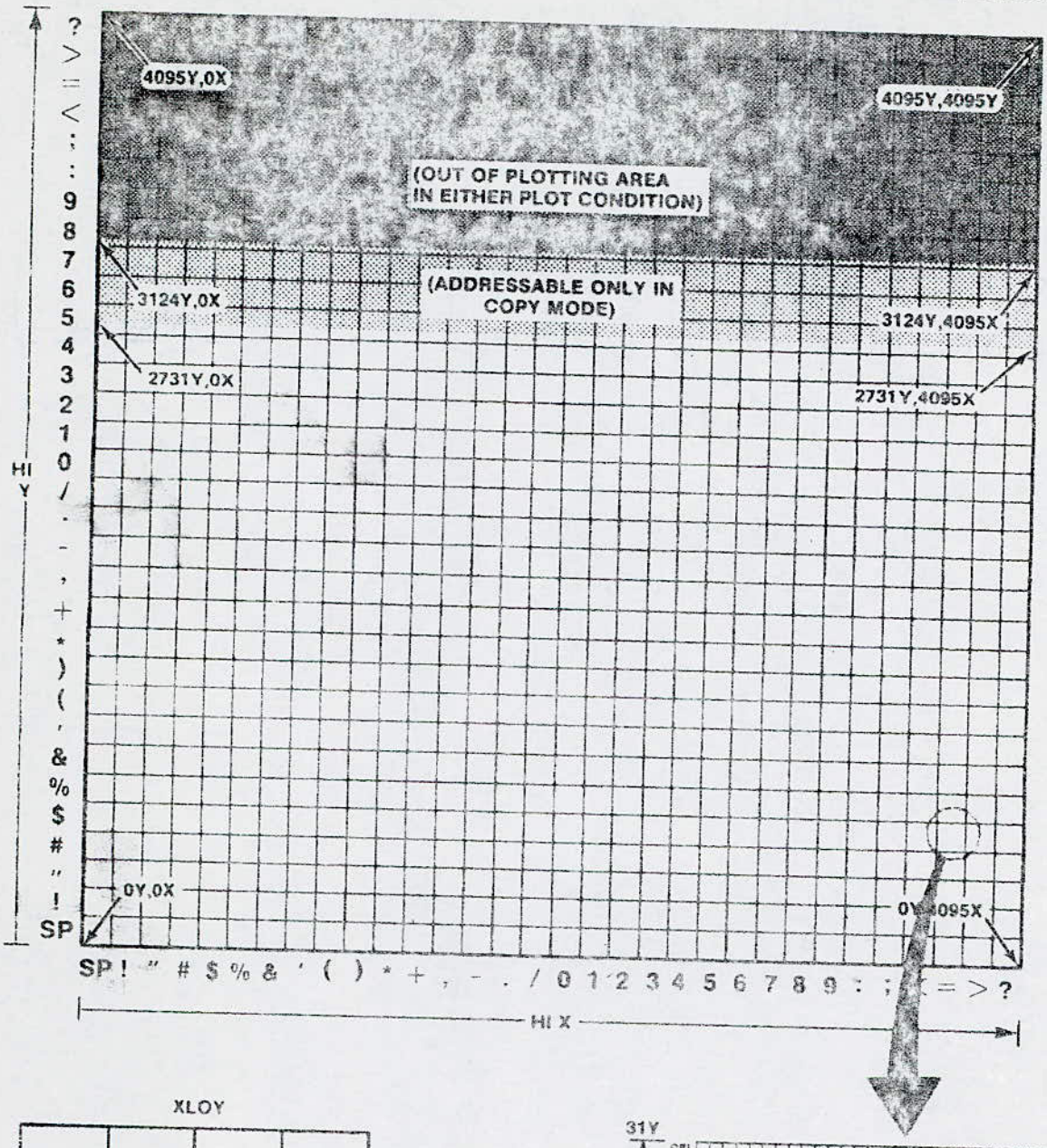
l'équipement 'Terminal' est prêt.

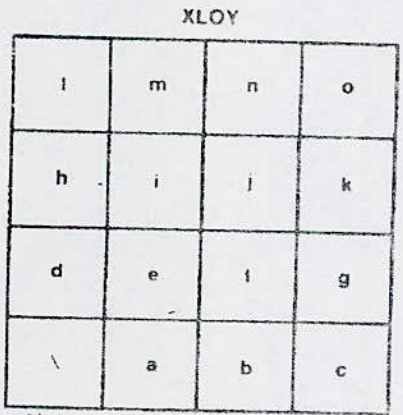
* DCD (Data Carrier detect)

ligne de détection de porteuse.

* Caractéristiques de la norme RS232

- Mode : 1 fil ;
- Longueur maximale du câble : 17 mètres ;
- Débit maximum : 20 K Bauds ;
- Tension minimale de sortie en charge : $\pm 0.5^V$ à $\pm 15^V$
- Tension maximale en circuit ouvert : $\pm 25^V$
- Résistance de sortie minimale : $R_0 = 300\ \Omega$
- Courant maximal de court circuit : ± 0.5 A
- Seuil maximal du recepteur : $+3^V$ à -3^V
- Tension maximale à l'inter recepteur : -25^V à 25^V





Note that characters from the second LOY column may also be used; these are shown for consistency.

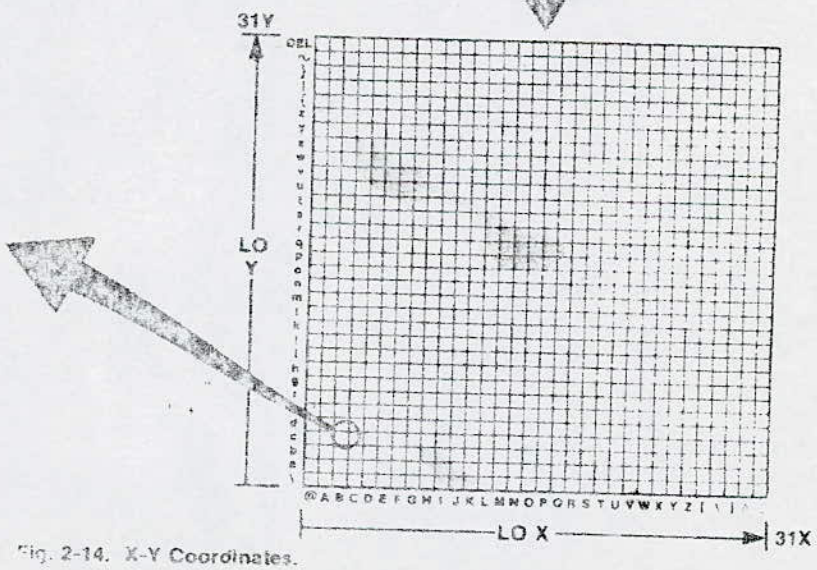


Fig. 2-14. X-Y Coordinates.

Caracteristiques de la table traçante

4 - Caractéristiques de la bibliothèque mathématique

Le format du nombre est compatible avec l'architecture du 6800 (24 bits : 16 pour la mantisse, 8 pour l'exposant)

Les chiffres négatifs sont complémentés à deux : les 16 bits de la mantisse en complément à deux et idem pour les 8 bits de l'exposant.

* la virgule du chiffre binaire en représentation virgule flottante apparaît juste après le bit de signe.

* les nombres normalisés, le MSByte de la mantisse détermine si le nombre est positif, négatif ou nul.

Quelques exemples de nombres

* 1 (décimal) s'écrit : 40001 en virgule flottante.

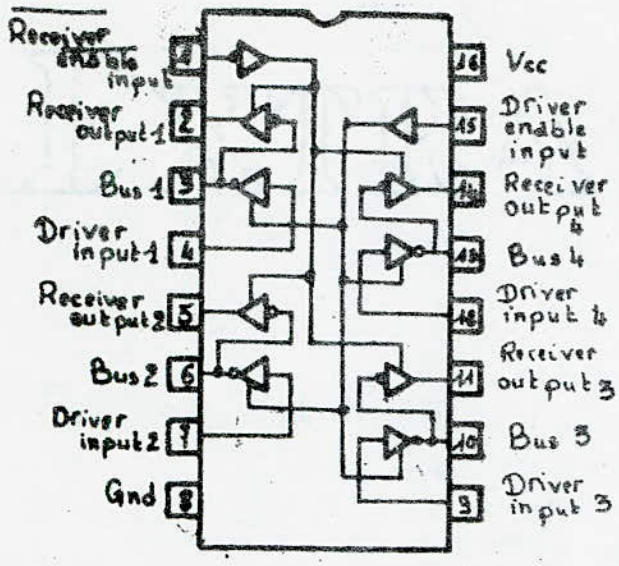
Car : $1 = \frac{1}{2} * 2^1$ et on a :

0.100	0000	0000	0000	0000	0001
4	0	0	0	0	1

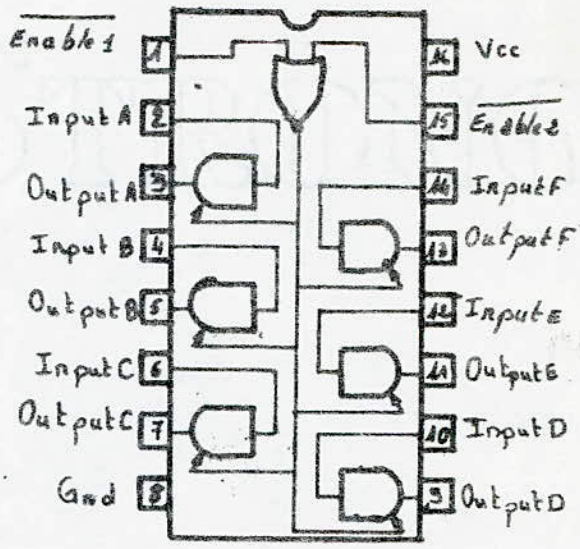
* 0.1 (décimal)

$0.1 \simeq 0.8 * 2^{-3}$

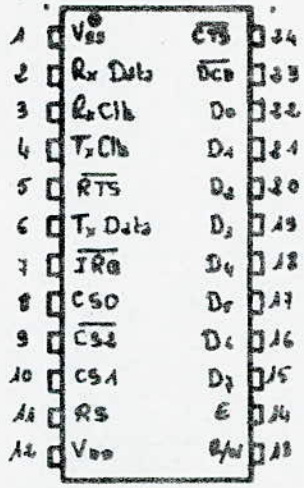
Il faut représenter (-3) en code complément à 2 d'où $(0,1)_{10}$ = 66 66 FD en virgule flottante.



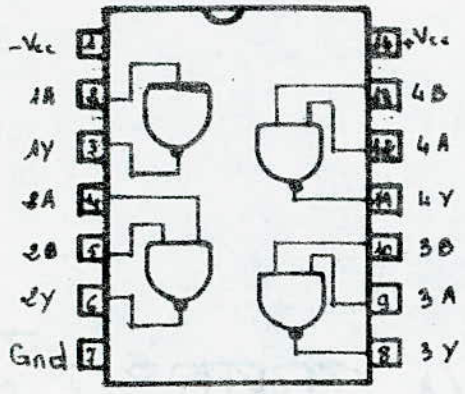
MC 8T26



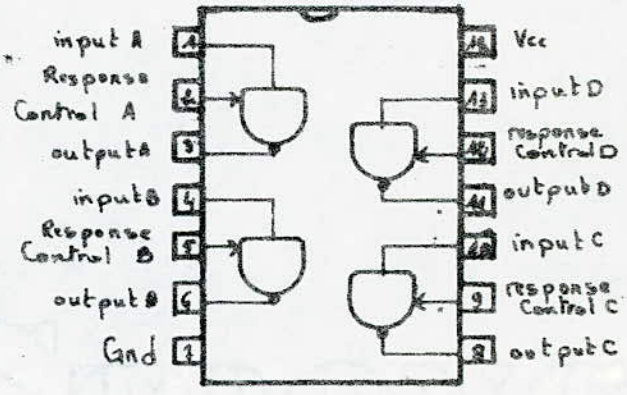
MC 8T95



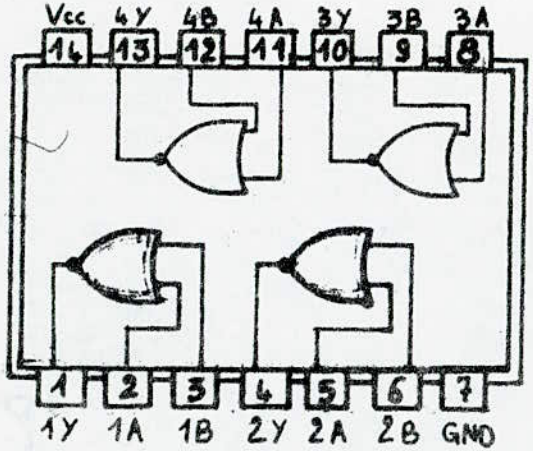
MC 6850



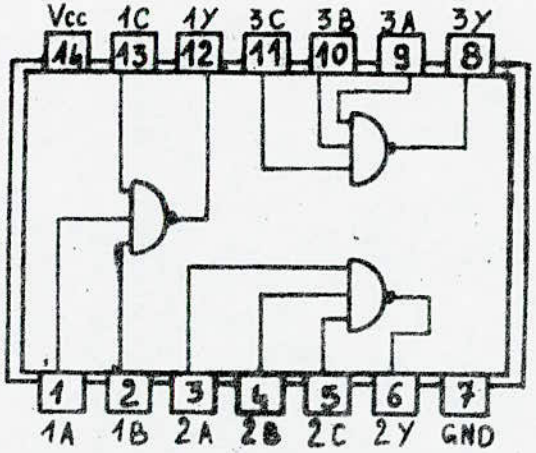
MC 1488



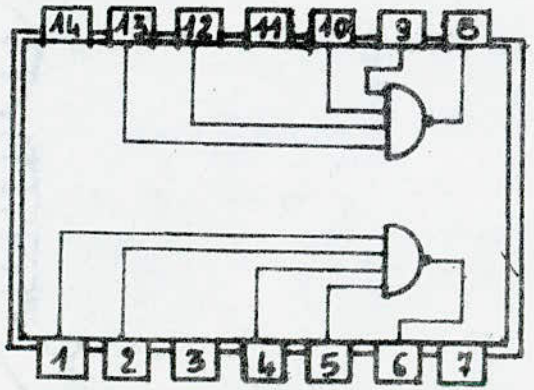
MC 1489



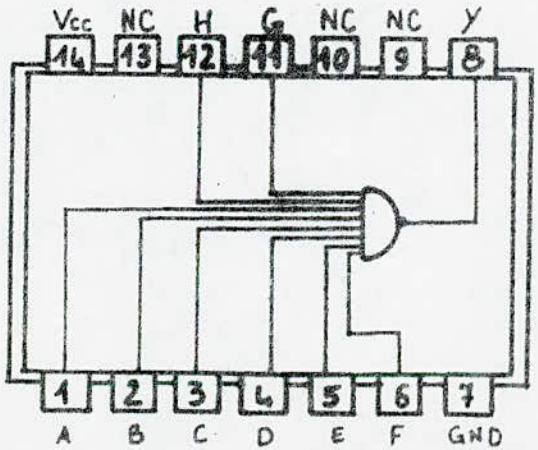
SN 7402



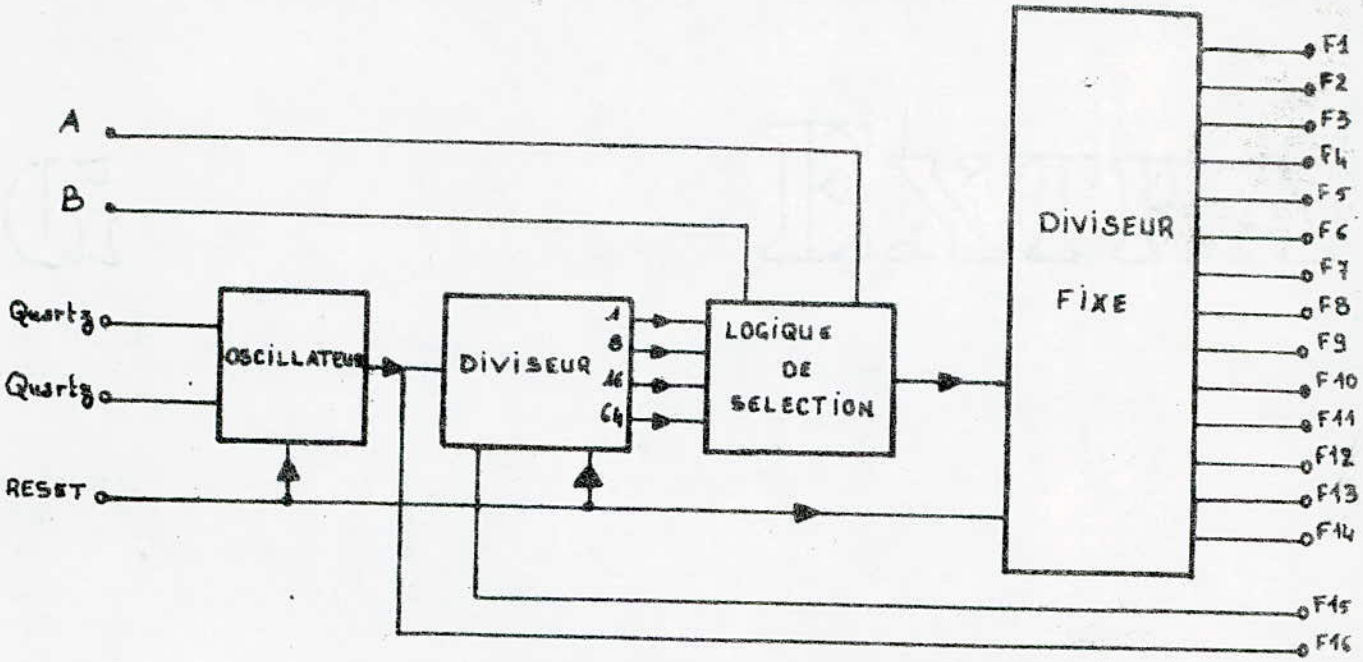
SN 7410



SN 7420



SN 7430

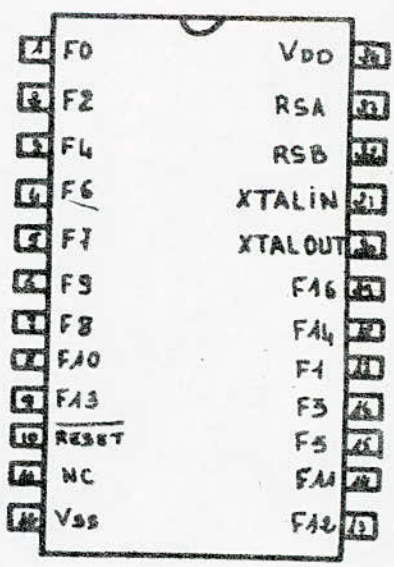


Synoptique du 14411

FREQUENCES				
Sortie	A=1 B=1	A=0 B=1	A=1 B=0	A=0 B=0
F1	614.4K	153.6K	76.8K	9600
F2	460.8K	115.2K	57.6K	7200
F3	307.2K	76.8K	38.4K	4800
F4	230.4K	57.6K	28.8K	3600
F5	153.6K	38.4K	19.2K	2400
F6	119.8K	29.9K	14.4K	1800
F7	76.8K	19.2K	9600	1200
F8	38.4K	9600	4800	600
F9	19.2K	4800	2400	300
F10	12.8K	3200	1600	200
F11	9600	2400	1200	150
F12	8613.2	2153.3	1076.6	134.5
F13	7085.5	1758.8	879.4	109.9
F14	4800	1200	600	75
F15	921.6K	921.6K	921.6K	921.6K
F16	1.843M	1.843M	1.843M	1.843M

M = MHz, K = KHz, rien = Hz

fréquences générées



brochage

BIBLIOGRAPHIE

[1]. B. Kubert, J. Szabo, and S. Guilieri

"The perspective représentation of functions of two variables"

Vol. 2, pp. 193-204, Avr. 1968

[2]. H Williamson

"Hidden line plotting program"

Commun. Ass. comput. Mach, vol 15,

pp. 100-103, Fev. 1972

[3]. T. Wright

"A two space solution to the hidden line problem for plotting functions of two variables."

IEE Transact. on comput, Vol C-22,

pp. 28-33, JAN 1973.

[4]. M. Lucas

"Laréalisation des logiciels graphiques interactifs"

Ed. Eyrolles. 1979

[5]. P. Morvan et M. Lucas

"Image et ordinateur, introduction à l'infographie interactive".

Ed. Larousse. Université, 1976

[6]. LANCE . A. LEVENTHAL

"6800. Programmation en langage assembleur."

Ed. Radio. 1982

[7]. M. Aumiaux

"Emploi des microprocesseurs"

Ed. Masson.

[8]. Documentation TEKTRONIX

* Interactive digital plotter 4662

* Data communication interface.

[9]. Revues "Haut parleur"

N° 1693 - Juin 1983, pp 71-76

N° 1694 - Juill 1984, pp 57-62

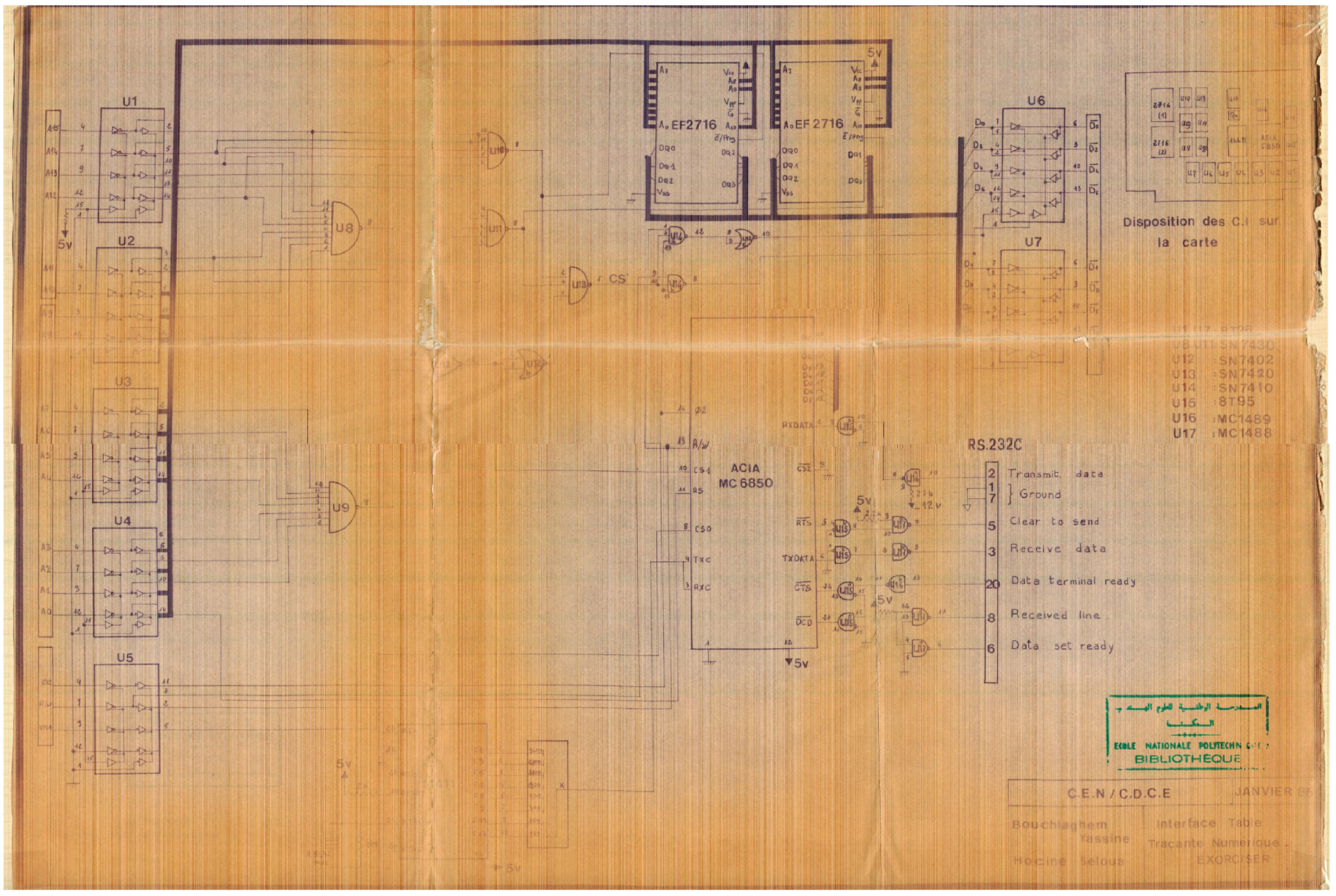
[10]. Projets de fin d'étude :

* "Résolution d'équations differentielles par microprocesseur"

JUIN 1981.

* "Introduction du microprocesseur dans le domaine de

la C.A.O" , Fevrier 1983.



C.E.N / C.D.C.E JANVIER 86

Bouchlaghem Yassine — Interface Table Tracante Numérique

Hocine Belouat — EXORCISER