

19 RÉPUBLIQUE FRANÇAISE  
INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
PARIS

11 N° de publication : **2 565 014**  
(à n'utiliser que pour les  
commandes de reproduction)  
21 N° d'enregistrement national : **85 07686**  
51 Int Cl<sup>a</sup> : G 09 G 1/16; G 11 C 15/00.

12

## DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 22 mai 1985.

30 Priorité : US, 23 mai 1984, n° 06/613 605.

43 Date de la mise à disposition du public de la  
demande : BOPI « Brevets » n° 48 du 29 novembre 1985.

60 Références à d'autres documents nationaux appa-  
rentés :

71 Demandeur(s) : Société dite : THE BOARD OF TRUS-  
TEES OF THE LELAND STANFORD JUNIOR UNIVERSITY,  
constituée selon les lois de l'Etat de Californie. — US.

72 Inventeur(s) : Stefan G. Demetrescu.

73 Titulaire(s) :

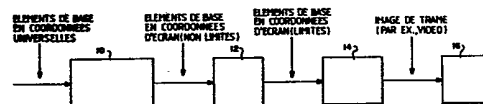
74 Mandataire(s) : Rinuy, Santarelli.

54 Système de mémoire rapide et procédé de traitement de données pour produire une trame d'éléments d'image, et  
segment de mémoire rapide.

57 L'invention concerne un système de mémoire rapide à  
architecture parallèle.

Ce système comporte une unité 10 de transformation géo-  
métrique qui reçoit des éléments de base correspondant à une  
information à afficher, un générateur 14 de visualisation avec  
balayage par trame, et un dispositif d'affichage 16. Les don-  
nées de pixels correspondant à plusieurs lignes de balayage  
sont enregistrées dans plusieurs segments de mémoire aux-  
quels on peut accéder en parallèle. Des bascules permettent  
une manipulation en continu des données enregistrées dans  
les segments de mémoire en même temps que des données  
de ligne de balayage sont extraites.

Domaine d'application : affichage avec balayage par trame,  
etc.



FR 2 565 014 - A1

D

L'invention concerne d'une manière générale les mémoires numériques, et plus particulièrement des systèmes de mémoires rapides convenant à la commande d'un visuel à trame et autre.

5           Brièvement décrit, un visuel à trame est un dispositif de sortie quelconque qui produit une image en édifiant sélectivement la couleur et/ou l'intensité de nombreux petits points (ou éléments d'image, également appelés pixels) qui sont agencés en un réseau rectangu-  
10           laire régulier. Un tel visuel peut comprendre des dispositifs à régénération périodique tels que le visuel à tube à rayons cathodiques ou des dispositifs d'impression sur papier tels que les imprimantes à laser à trame xérogra-  
                  phique.

15           La création d'image par ordinateur s'est éloignée des visuels calligraphiques, tels que les tubes à rayons cathodiques à balayage cavalier et les tables traçantes pour se rapprocher des visuels à trame tels que les écrans de télévision et les imprimantes matricielles par  
20           pages. Cette conversion est due à de nombreuses raisons parmi lesquelles: (1) les visuels à trame coûtent beaucoup moins cher que d'autres procédés de visualisation; (2) la plupart des visuels à trame reposent sur une mémoire tampon de séquence et le coût d'une mémoire à semi-  
25           conducteur a récemment décliné de façon brusque; (3) les visuels à trame peuvent remplir des surfaces de couleurs pleines (et d'ombre), alors que les visuels télégraphiques ne peuvent dessiner efficacement que des contours; et (4) les visuels à trame peuvent afficher des caractères de nombreuses formes de polices, de façon plus  
30           naturelle et plus efficace que les visuels calligraphiques.

                  Un système d'affichage graphique typique présente des descriptions de haut niveau d'une image à deux ou trois dimensions, en coordonnées universelles qui sont  
35           les coordonnées décrivant le plus naturellement l'image.

Cette image est transformée et limitée à l'aide de procédés graphiques bien connus en une représentation à deux dimensions sous la forme d'éléments graphiques de base décrits dans les coordonnées de l'écran d'affichage.

5 Ces fonctions de transformation ont été incorporées dans un circuit intégré à très grande échelle, de conception VLSI, comme décrit dans le brevet des Etats-Unis d'Amérique n° A-36257. Un convertisseur ou générateur de visualisation avec balayage par trame ajoute ces éléments de  
10 base transformés à l'image formée par balayage par trame, partiellement achevée (c'est-à-dire qu'il modifie l'intensité de certains des pixels de la trame ou du réseau de l'image) et il affiche ou imprime également la trame de l'image.

15 Malheureusement, le passage des visuels calligraphiques aux visuels à balayage par trame a posé de nouveaux problèmes. Dans un système à trame, il est nécessaire, non seulement de calculer les positions des éléments de base, mais également de remplir tous les  
20 éléments d'image à l'intérieur des éléments de base à l'aide des valeurs souhaitées. Actuellement, la vitesse à laquelle des polygones peuvent être remplis est généralement très inférieure à la vitesse à laquelle la position des polygones peut être calculée.

25 En conséquence, l'utilisation de visuels à trame pour des images produites en temps réel est limitée et coûteuse. Par exemple, si une image de 1000 par 1000 pixels doit être redessinée 30 fois par seconde, il faut généralement accéder à plus de 30 millions de pixels par  
30 seconde. Ceci pose le problème de la visualisation avec balayage par trame.

Conformément à l'invention, il est prévu un système de mémoire rapide et de processeurs qui comprend plusieurs segments de mémoire. Chaque segment de mémoire  
35 comprend un réseau de mémoires vives et un processeur qui

commande l'enregistrement, l'accès et la manipulation des données dans le réseau. Plusieurs segments de mémoire coopèrent de façon à enregistrer des données de pixel correspondant à plusieurs lignes de balayage par trame  
5 mais travaillent en réponse à un processeur de ligne de balayage partagé. Le processeur de ligne de balayage reçoit des données transformées et limitées d'un processeur de transformation graphique et de limitation et convertit chaque objet graphique qui lui est transmis en  
10 une séquence de segments de pixels horizontaux qui sont présentés aux segments de mémoire, en tant qu'ordres de la forme: ligne de balayage (Y), point de départ ( $X_s$ ), point de fin ( $x_e$ ), configuration de remplissage de pixels et opérations de l'unité logique arithmétique. Chaque  
15 processeur de segment de mémoire réagit à ces ordres de segments horizontaux en mettant à jour les segments de mémoire en réponse auxdits ordres.

L'invention a donc pour objet un système de mémoire rapide, et notamment un système de mémoire  
20 comprenant plusieurs segments de mémoire commandés chacun par un processeur exclusif. L'invention a également pour objet un système de mémoire hautement associatif qui est aisément réalisé à l'aide des techniques d'intégration à très grande échelle.

25 L'invention sera décrite plus en détail en regard des dessins annexés à titre d'exemples nullement limitatifs et sur lesquels :

. la figure 1 est un schéma fonctionnel simplifié d'un système d'affichage graphique;

. la figure 2 est un schéma fonctionnel simplifié du générateur de visualisation avec balayage par  
30 trame de la figure 1, comprenant un système de mémoire conforme à l'invention;

. la figure 3 est un schéma fonctionnel simplifié d'un système de mémoire comprenant plusieurs segments  
35 de mémoire conformes à l'invention et tels qu'utilisés

dans le générateur de visualisation avec balayage par trame de la figure 2;

. la figure 4 est un schéma fonctionnel simplifié du processeur de lignes de balayage de la figure 3;

5 . la figure 5 représente un polygone à afficher et illustre le fonctionnement du processeur de lignes de balayage;

10 . la figure 6 illustre l'effet de chacun des ordres de remplissage de lignes horizontales transmis par les processeurs de lignes de balayage aux segments de mémoire;

. la figure 7 est un schéma fonctionnel simplifié d'un segment de mémoire conforme à l'invention;

15 . la figure 8 est un schéma fonctionnel simplifié d'une unité logique arithmétique (ULA) de lignes de balayage dans les segments de mémoire de la figure 7;

. les figures 9 à 11 sont des schémas fonctionnels simplifiés de variantes d'agencements de systèmes de mémoire conformes à l'invention;

20 . la figure 12 est un schéma fonctionnel simplifié d'un segment de mémoire qui permet un nuancement doux;

25 . la figure 13 est un schéma d'une structure arborescente de multiplicateurs convenant au segment de mémoire de la figure 12; et

. la figure 14 est un schéma d'un réseau général d'unité logique arithmétique et de circuits associés, conformes à l'invention.

30 En référence à présent aux dessins, la figure 1 est un schéma fonctionnel simplifié d'un système d'affichage graphique dans lequel des éléments de base, en coordonnées universelles (par exemple un polygone ou une ligne), sont transformés, comme indiqué en 10, en coordonnées d'écran qui sont ensuite limitées en 12, pour  
35 la commande d'un dispositif d'affichage ou visuel.

Les fonctions des unités 10 et 12 peuvent être réalisées par une machine à géométrie comme décrit dans le brevet n° A-36257 précité. Les coordonnées, telles que transformées et limitées pour être utilisées dans le visuel, sont  
5 ensuite appliquées à un générateur de visualisation avec balayage par trame 14 qui comprend une mémoire de masse destinée à enregistrer l'image partiellement construite sous la forme d'un réseau de pixels et des moyens destinés à commander les lignes de balayage par trame dans le  
10 dispositif d'affichage 16.

Comme décrit précédemment, le dispositif d'affichage ou visuel peut comprendre une image de 1000 par 1000 pixels qui doit être redessinée 30 fois par seconde sur un tube à rayons cathodiques. Par conséquent, il faut  
15 accéder à chaque seconde à des données correspondant à 30 millions de pixels.

En variante, le visuel peut être une imprimante matricielle ou à trame capable d'imprimer, à chaque seconde, une feuille de papier de 21,5 par 28 cm. Si la résolution  
20 est de 118 pixels par cm dans les directions X et Y, il faut accéder à chaque seconde à 8,4 millions de pixels.

La figure 2 est un schéma fonctionnel simplifié d'un générateur de visualisation avec balayage par trame utilisant un système de mémoire rapide conforme à  
25 l'invention. Le générateur comprend un processeur 20 de lignes de balayage, plusieurs segments de mémoire qui sont commandés par le processeur 20, et un contrôleur 24 d'affichage. Le processeur 20 de lignes de balayage reçoit des éléments de base en coordonnées d'écran du  
30 processeur 10 de transformation géométrique, et le processeur 20 produit alors des ordres de remplissage d'une ligne horizontale RLH  $(Y, X_s, X_e)$  qu'il transmet aux segments 22 de mémoire. Les données provenant des segments de mémoire sont ensuite produites sous forme  
35 numérique pour une image de trame qui est transmise au

contrôleur 24 du visuel afin de commander ce dernier.  
Le processeur de lignes de balayage convertit chaque  
élément de base graphique en séquences de pixels hori-  
zontaux à remplir, comme décrit plus en détail ci-après  
5 en référence aux figures 4 et 5. Les segments 22 de  
mémoire ont pour fonction de maintenir l'image de trame  
(c'est-à-dire le réseau de pixels) et de la modifier à  
la réception d'ordres de remplissage d'une ligne horizon-  
tale du processeur de lignes de balayage. La fonction  
10 exacte des ordres de remplissage de ligne horizontale  
sera décrite ci-après en référence aux figures 4 et 6.  
Le contrôleur 24 du visuel extrait l'image tramée des  
processeurs de trame et commande le visuel à trame ou  
l'imprimante à trame.

15 La figure 3 est un schéma fonctionnel simplifié  
d'un système de mémoire comprenant plusieurs segments de  
mémoire conformes à l'invention et tel qu'utilisé dans le  
générateur de visualisation avec balayage par trame de la  
figure 2. Dans cette forme de réalisation, 16 processeurs  
20 de lignes de balayage commandent un réseau de 64 seg-  
ments 22 de mémoire qui commandent les données de pixels  
pour un visuel comportant 24 lignes de balayage avec  
1024 pixels par ligne de balayage. Dans cette forme de  
réalisation, chaque processeur de ligne de balayage  
25 commande quatre segments de mémoire qui, ensemble, enre-  
gistrent et modifient les données pour 64 lignes de  
1024 pixels chacune. Chaque segment de mémoire peut  
comprendre une mémoire à 16K avancée en 64 lignes avec  
256 bits de données par ligne. Chaque groupe de segments  
30 de mémoire travaille sous la commande de l'un des 16  
processeurs 20 de lignes de balayage, ce qui permet des  
opérations indépendantes et en parallèle des groupes  
de segments de mémoire. En outre, chaque segment 22 de  
mémoire comprend son propre processeur de manière à  
35 pouvoir être manipulé en parallèle avec d'autres segments

de mémoire commandés par le processeur partagé 20 de lignes de balayage.

La figure 4 est un schéma fonctionnel simplifié d'une forme préférée de réalisation du processeur de ligne de balayage. Pour plus de clarté, le processeur ne traite que des caractères et des polygones monotones dans lesquels une ligne horizontale coupe au maximum deux fois le périmètre du polygone. La figure 5 illustre un tel polygone. Les sommets du polygone sont présentés au processeur en ordre Y décroissant, et chaque sommet est identifié selon qu'il fait partie du bord gauche ou du bord droit du polygone.

En référence à la figure 4, des ordres présents sur le bus 30 sont interprétés par le décodeur 32 d'ordre qui a pour fonction de distribuer les ordres vers le bloc fonctionnel parallèle de mémoire approprié. Chaque bloc de fonction parallèle est composé d'un ordinateur à programme mémorisé classique, ainsi qu'il est bien connu dans la technique.

Le décodeur 32 d'ordre peut recevoir quatre types généraux d'ordres: (i) remplissage de mémoire 40 d'image tramée avec une configuration donnée qui sera utilisée pour remplir l'intérieur des polygones suivants, (ii) remplissage d'une mémoire 42 de police qui sera utilisée pour placer ensuite des caractères dans l'image formée par trame, (iii) visualisation avec balayage par trame du polygone en validant le processeur 34 de polygone, (v) visualisation avec balayage par trame d'un caractère en validant le processeur 44 de police.

Le processeur 34 de polygone a pour fonction de visualiser par balayage par trame le polygone en cours jusqu'à l'extrémité du bord droit ou gauche réel. A ce moment, le processeur 34 attend le bord suivant du décodeur d'ordre. Lorsque le bord suivant est reçu, la visualisation avec balayage par trame du polygone se

poursuit au moyen d'algorithmes de ligne de balayage bien connus dans la technique. Les deux processeurs 36 et 38 de bords calculent simultanément les coordonnées X de commencement et de fin de la ligne suivante de balayage à visualiser par trame, à l'aide de procédés bien connus dans la technique. La figure 5 illustre ces opérations.

Après que les deux processeurs de bords ont calculé l'intersection de la ligne de balayage en cours (Y) avec les deux bords du polygone, cette information est transmise aux segments de mémoire sous la forme d'un ordre de remplissage de ligne horizontale comprenant: (i) une coordonnée Y (c'est-à-dire la ligne de balayage) devant être modifiée, (ii) le premier pixel devant être affecté (qui a été calculé par le processeur de bord gauche 38), (iii) le dernier pixel devant être affecté (qui a été calculé par le processeur de bord droit), et (iv) la combinaison tramée à 16 bits à utiliser en combinaison répétée pour remplir le segment horizontal choisi. L'effet de cet ordre est illustré sur la figure 6.

La combinaison de trame est choisie par le processeur 34 de polygone à partir de l'une des 16 combinaisons enregistrées dans la mémoire 30 de trame. Ces combinaisons y sont enregistrées à l'aide d'ordres transmis au processeur 20 de ligne de balayage par l'intermédiaire du bus 30 de ce processeur. Le processeur 34 de polygone choisit l'une des 16 combinaisons en utilisant la fonction [(coordonnée Y actuelle) module 16]. Ceci a pour effet de répéter la combinaison de trame toutes les 16 lignes de balayage.

Le processeur 44 de police a pour fonction de placer le caractère en cours dans la trame. Il extrait la combinaison de caractère de la mémoire de police et utilise le dispositif 46 de décalage de tambour pour aligner convenablement la combinaison de caractère à

mettre en place dans les segments de mémoire. Chaque caractère est placé dans la trame d'image d'un grand nombre de sections horizontales à 16 bits par l'émission d'ordres de remplissage de ligne horizontale comme montré sur la figure 6, lesquels ordres ne modifient que 16 pixels à la fois, et à l'aide d'une combinaison de trame qui représente une des lignes de balayage du caractère devant être visualisé avec balayage par trame. Ainsi, chaque caractère est visualisé avec balayage par trame par l'émission d'un ordre de remplissage de ligne horizontale pour chaque ligne de balayage que le caractère occupe.

Il convient de noter que toutes les fonctions du processeur de ligne de balayage peuvent être exécutées par un ordinateur à programme enregistré classique (par exemple un microprocesseur "Motorola 68000" avec mémoire associée), programmé au moyen d'algorithmes permettant d'exécuter les opérations décrites et qui sont bien connus dans la technique. La forme préférée de réalisation décrite ci-dessus accélère simplement la fonction du processeur de ligne de balayage par la mise en oeuvre en parallèle de plusieurs processeurs classiques pour parvenir au même résultat.

La figure 7 est un schéma fonctionnel simplifié d'un segment de mémoire d'une forme de réalisation de l'invention, composé de 6 sections principales.

La mémoire principale 50 est une mémoire vive statique ou dynamique normale. Il est souhaitable d'utiliser un réseau beaucoup plus large que long afin de pouvoir travailler le plus possible en parallèle. Dans cette forme de réalisation, une mémoire vive à 16K bits doit être utilisée, cette mémoire étant organisée en 64 mots (c'est-à-dire rangées) de 256 bits (c'est-à-dire colonnes) chacun.

L'unité logique et arithmétique 52 de trame

(ULA) intercepte la combinaison arrivante de trame à 16 bits et exécute de simples opérations booléennes qui permettent une représentation tramée de plusieurs valeurs tout en formant des images d'éléments de base.

5 La combinaison de trame arrivante peut être interprétée de l'une des quatre manières suivantes: (1) elle est utilisée telle quelle; (2) elle est inversée bit par bit avant d'être utilisée; (3) elle est ignorée et tous les bits 1 sont alors utilisés, (4) elle est ignorée et tous

10 les bits 0 sont alors utilisés. Ceci permet de former une image tramée de valeurs multiples tout en formant des images d'éléments de base. Si chacun des pixels peut prendre l'un de 8 niveaux de gris, il est possible de former une image tramée au moyen d'un mélange de deux

15 des 8 valeurs de l'échelle des gris. Par exemple, pour obtenir une intensité de 5,5, on peut remplir un polygone d'une configuration alternée de valeurs de gris 5 et 6. Cet effet peut être obtenu par la transmission d'ordres de remplissage de pixels au processeur de

20 segments de mémoire tout en ordonnant que le plan des bits de poids fort utilise une combinaison de trame constituée de tous les bits 1, le plan médian utilise la combinaison de trame telle que fournie, et le plan de poids faible utilise la combinaison inversée. Ceci a

25 pour effet de placer un 6 dans tous les emplacements où la combinaison de trame est 1 et 5 ailleurs.

Le comparateur 54 en parallèle établit tous les bits de sortie dont la position est inférieure à la coordonnée X donnée. Ceci a pour effet de choisir les

30 limites gauche et droite des pixels devant être affectés lors de l'exécution d'un ordre de remplissage de ligne horizontale. Ces limites sont utilisées par l'unité logique et arithmétique 56 des lignes de balayage.

L'unité ULA 56 de ligne de balayage détermine

35 la valeur devant être réenregistrée dans le réseau de

mémoire, en fonction des valeurs d'entrée provenant du comparateur en parallèle 54, de l'unité ULA 52 de trame (par l'intermédiaire du bus de trame) et du réseau de mémoire 50.

5 Les bascules 58 d'affichage bloquent une ligne de balayage provenant d'amplificateurs différentiels 60 afin que la ligne puisse être retirée des segments de mémoire indépendamment du fonctionnement de la partie restante des composants des segments de mémoire.

10 Une logique 62 commande le réseau de mémoire, le comparateur en parallèle, les unités ULA et les bascules d'affichage afin de leur faire exécuter les ordres de remplissage de ligne horizontale que cette commande reçoit.

15 Pour distribuer les bits répétés de combinaisons de trame vers les bits correspondants des mots de mémoire, chacun des 16 bits provenant de l'unité ULA de trame est transmis à chaque seizième colonne. Ceci est réalisé au moyen d'un bus 34 de 16 bits s'étendant  
20 horizontalement au-dessus du réseau de mémoire. Si l'on souhaite placer des combinaisons qui sont alignées par rapport aux coordonnées X de début (par exemple pour des caractères à visualiser par balayage par trame), il est nécessaire de faire tourner la combinaison de  $X \bmod 16$ .  
25 Cette rotation peut être exécutée par le processeur de ligne de balayage sans accroissement de la bande passante entre le processeur de ligne de balayage et le segment de mémoire.

30 La figure 8 est un schéma fonctionnel simplifié de l'unité ULA 56 de ligne de balayage et on décrira ci-après un cycle classique de fonctionnement de cette unité lors de l'exécution d'une opération de remplissage de ligne horizontale.

35 En premier lieu, la coordonnée de début (incluse) (Xs) de l'étendue X (c'est-à-dire étendue d'une colonne) à

affecter est présentée au comparateur en parallèle et l'inverse de son signal de sortie est bloqué dans une bascule L1. Ainsi, la bascule L1 est dans l'état vrai pour tous les emplacements (c'est-à-dire colonnes) situés le long de la ligne de balayage, qui sont supérieurs ou égaux à  $X_s$ .

En second lieu, la coordonnée de fin (exclusive) ( $X_e$ ) de l'étendue  $X$  est présentée au comparateur en parallèle et le signal de sortie de celui-ci est bloqué dans une bascule L2. Par conséquent, la bascule L2 est dans l'état vrai pour tous les emplacements situés le long de la ligne de balayage, qui sont inférieurs à  $X_e$ . Par conséquent,  $SEL(j)$  est vrai pour tous les  $X$  dans l'intervalle  $(X_s, X_e)$ .

A ce moment, le réseau de mémoire vive a récupéré les valeurs actuelles des pixels ( $IR(j)$ ) dans la ligne de balayage alors choisie. L'unité ULA travaille sur les bits choisis comme souhaité et génère les pixels  $IW(j)$  devant être réécrits dans la mémoire.

Pour que l'unité ULA reste aussi simple que possible, seul le groupe minimal d'opérations suivantes doit être exécuté: (i) aucune opération, effectuer ( $IW(j) = IR(j)$ ), (ii) remplacer les pixels de trame dans tous les emplacements choisis de pixels, (iii) soumettre à une opération OU les pixels de trame avec tous les pixels choisis. D'autres fonctions sont possibles (par exemple toutes les opérations booléennes connues) au prix d'un agrandissement de l'unité ULA.

Les figures 9 à 11 sont des schémas fonctionnels simplifiés de variantes de systèmes de mémoire conformes à l'invention. Sur la figure 9, chaque processeur de ligne de balayage commande deux rangées de segments de mémoire, réduisant ainsi le coût du système de mémoire, mais réduisant également les opérations effectuées en parallèle. Sur la figure 10, il est prévu un système à double

tampon dans lequel un premier jeu de segments de mémoire est affiché tandis qu'un autre jeu est commandé par les processeurs de ligne de balayage qui génèrent la trame suivante à afficher. Cet agencement permet au processeur  
5 de ligne de balayage d'être complètement utilisé. La figure 11 représente un système de mémoire à bits multiples par pixel (par exemple une échelle de gris). Pour la commande de plans de bits multiples, il suffit d'ajouter deux lignes indépendantes de commande entre chaque  
10 processeur de ligne de balayage et chaque plan de bit séparé. L'ensemble des lignes de commande peut encore être partagé entre tous les segments de mémoire dans tous les plans de bits.

Au prix d'une certaine complication de l'architecture des segments de mémoire, il est possible d'ajouter  
15 une possibilité de nuancement douce de Gouraud comme montré sur la figure 12. Chaque pixel est enregistré sous la forme d'une valeur d'intensité de K bits "verticalement" le long d'une colonne du réseau de  
20 mémoire, comme montré. La sous-gamme appropriée de pixels X peut être calculée par le comparateur en parallèle, comme précédemment. Cependant, étant donné que les pixels sont enregistrés verticalement, au moins K cycles de mémoire sont nécessaires pour enregistrer des intensités dans  
25 les pixels choisis.

Pour nuancer en douceur un polygone (opération connue sous le nom de nuancement de Gouraud dans la technique), il est nécessaire de placer les valeurs d'intensité en interpolation linéaire à chaque pixel le long  
30 d'une ligne de balayage. Il est heureusement aisé de générer une interpolation linéaire en série de bits au moyen d'une structure binaire arborescente, similaire à un multiplicateur en série tel qu'illustré sur la figure 13. Chaque noeud de la structure arborescente  
35 est constitué soit par un simple additionneur série,

soit par une unité de retard. Lorsque le coefficient A et la constante C sont insérés en série dans la structure arborescente (par le processeur de ligne de balayage), chaque noeud de feuille de la structure arborescente commence à générer un bit de la valeur  $Ax + \text{constante}$ , où x représente la position physique de la feuille dans la structure arborescente telle que montrée. Si les intensités doivent être comprises avec précision dans une valeur d'intensité, A doit être représenté sous la forme d'un nombre de points fixes, avec une partie fractionnaire de dimension égale au nombre total de bits demandé pour représenter la coordonnée X maximale (appelée N) (par exemple si une intensité de 8 bits est souhaitée pour un écran d'une largeur de 1024 pixels, A doit avoir 8 bits entiers et 10 bits fractionnaires).

En conséquence, chaque ligne de balayage d'un polygone à nuancement doux demande  $N + K$  cycles de processeur dont seulement les K derniers enregistrent des bits dans les pixels choisis. Cependant, pour une représentation de la totalité des K bits par pixel, on doit à présent réaliser un système comportant K fois plus de processeurs (que pour un système à un bit par pixel) pour tenir compte des bits supplémentaires. Ceux-ci peuvent tous travailler en parallèle de sorte que la diminution effective de performance (par rapport à un système à un bit par pixel) n'est que  $(N + K) / K$ , qui est environ 2 si N et K sont approximativement égaux. Par conséquent, il faut seulement deux fois plus de temps pour remplir un polygone à nuancement doux que pour remplir un polygone à intensité constante.

Jusqu'à présent, on a décrit un système à fonction particulière, qui est adapté à la fonction particulière d'une visualisation avec balayage par trame rapide. Cependant, en généralisant légèrement l'unité ULA et le circuit associé (le trajet des données)

pour le bord supérieur du réseau de mémoire, comme montré sur la figure 14, on obtient un trajet général de données de processeurs hautement parallèle, à usage général, pouvant être programmé pour l'exécution de nombreuses tâches. L'entrée vers le réseau et la sortie du réseau peuvent être commandées au moyen d'un registre à décalage ou par tout autre dispositif extérieur. En raison de sa nature à usage général, il n'est pas possible de décrire toutes les utilisations particulières dans lesquelles une telle architecture peut être placée. Il est évident que l'une des utilisations est l'exécution d'une visualisation avec balayage par trame, mais toute tâche pouvant utiliser cette architecture peut être réalisée.

Ainsi qu'on peut le voir, la structure du processeur est similaire à celle d'un trajet de données d'ordinateur classique. La nouveauté réside dans le fait que (i) le processeur est associé à un grand réseau bidimensionnel de mémoire, auquel on peut accéder rangée par rangée, et (ii) le nombre de bits contenus dans le "mot" du trajet de données est très supérieur aux nombres utilisés dans la technique (256 ou plus au lieu de 16 ou 32), et (iii) en raison de ce "mot" élargi, le processeur et la mémoire sont placés physiquement à proximité l'un de l'autre, sur un seul circuit intégré.

Cette architecture ne serait pas possible s'il n'existait le rapprochement intime du trajet de données du processeur et de la mémoire sur laquelle il travaille en raison de l'impossibilité de connecter 256 (ou plus) mots de bits entre la mémoire et les unités de calcul lorsqu'elles sont physiquement séparées.

On a décrit un système de mémoire rapide qui est particulièrement avantageux pour la commande d'une image à trame. En utilisant plusieurs segments de mémoire ayant chacun son propre processeur, on obtient un fonctionnement en parallèle dans lequel la remise à jour et

la manipulation rapide des données sont facilitées. Les segments de mémoire favorisent en eux-mêmes l'application de techniques de fabrication de micro-circuits à intégration à très grande échelle (VLSI).

5                    Il va de soi que de nombreuses modifications peuvent être apportées au système décrit et représenté sans sortir du cadre de l'invention. Par exemple, un ordinateur classique à programme enregistré, basé sur le type AMD 2901, par exemple, peut être utilisé comme  
10                    processeur de ligne de balayage. Ce processeur peut être programmé à l'aide d'algorithmes graphiques connus dans la technique pour générer les ordres nécessaires aux segments de mémoire.

REVENDEICATIONS

1. Système de mémoire rapide pour former une trame d'éléments pixels constituant une image qui est affichée en réponse à des données de pixels, l'image étant réalisée par balayage suivant plusieurs lignes, le système de mémoire étant caractérisé en ce qu'il comporte un bus pour la circulation de données transformées d'éléments de base graphiques en coordonnées d'affichage, plusieurs processeurs (20) de lignes de balayage connectés de façon à recevoir les données transformées dudit bus, chaque processeur de ligne de balayage commandant des données pour plusieurs lignes de balayage, et plusieurs moyens de mémorisation de données destinés à enregistrer des données pour toutes les lignes de balayage, chacun des moyens de mémorisation de données étant connecté de façon à recevoir des données de l'un des processeurs de lignes de balayage, chacun desdits moyens de mémorisation de données comprenant plusieurs segments de mémoire (22) enregistrant chacun des données correspondant à une partie limitée de chacune des lignes de balayage commandées par le processeur de ligne de balayage auquel les moyens de mémorisation de données sont connectés, de façon à permettre un enregistrement, un accès et un travail en parallèle sur les données enregistrées.

2. Système de mémoire selon la revendication 1, caractérisé en ce que chaque segment de mémoire comprend un réseau (50) de mémoire vive représentant une partie d'une image de trame et un processeur qui, en réponse à des données correspondant à un numéro de ligne de balayage, à un point de départ et à un point de fin provenant dudit processeur de ligne de balayage, modifie des données choisies dans ledit réseau de mémoire vive.

3. Système de mémoire selon la revendication 2, caractérisé en ce que le processeur réagit en outre à

des données de combinaison de trame provenant du proces-  
seur de ligne de balayage, en enregistrant des données,  
et en y accédant, dans le réseau de mémoire vive.

4. Système de mémoire selon la revendication 2,  
5 caractérisé en ce que le processeur réagit en outre à  
des ordres provenant du processeur de ligne de balayage  
en exécutant des opérations de logique booléenne sur des  
parties choisies de données de trame enregistrées.

5. Système de mémoire selon la revendication 2,  
10 caractérisé en ce que le processeur réagit à des ordres  
comprenant la rangée du réseau de mémoire, les premier  
et dernier éléments d'un sous-ensemble contigu qu'une  
unité logique et arithmétique (56) doit modifier, la  
configuration de bits pour la configuration de trame, et  
15 l'opération de logique booléenne que l'unité logique et  
arithmétique doit exécuter.

6. Système de mémoire selon l'une des revendi-  
cations 2 et 5, caractérisé en ce que chaque segment de  
mémoire comprend une bascule (58) d'affichage destinée à  
20 recevoir des données des moyens à mémoire vive afin de  
les utiliser pour extraire l'image dudit segment de  
mémoire.

7. Système de mémoire selon la revendication 1,  
caractérisé en ce que les moyens à mémoires de données  
25 enregistrent des données de pixels sous forme de valeurs  
d'intensité.

8. Système de mémoire selon la revendication 7,  
caractérisé en ce que les moyens à mémoires de données  
comprennent plusieurs (K) positions d'enregistrement  
30 pour chaque valeur d'intensité de pixel (K bits), ces  
positions d'enregistrement étant agencées de manière  
qu'un bit de chacun des pixels d'une ligne de balayage  
soit accessible en même temps pour tous les pixels.

9. Système de mémoire selon la revendication 8,  
35 caractérisé en ce que chaque rangée du réseau de mémoire

contient l'un des grands K bits d'intensité d'une rangée de pixels.

10. Système de mémoire selon l'une des revendications 8 et 9, caractérisé en ce qu'il comporte en outre  
5 une structure arborescente binaire pour produire des valeurs d'intensité de pixels interpolées.

11. Système de mémoire selon la revendication 10, caractérisé en ce que les valeurs interpolées sont calculées simultanément pour tous les pixels choisis sur une  
10 ligne de balayage, lesdites valeurs étant produites pour tous les pixels choisis, un bit à la fois.

12. Procédé de traitement de données pour générer une image de trame à lignes multiples, caractérisé en ce qu'il consiste à enregistrer des données de pixels  
15 pour plusieurs lignes de balayage, dans des emplacements situés dans plusieurs segments de mémoire et auxquels on peut accéder en parallèle, et, simultanément, à accéder, traiter et modifier plusieurs emplacements de données de pixels pour toute ligne de balayage.

13. Procédé selon la revendication 12, caractérisé en ce qu'il consiste à bloquer des données de  
20 pixels provenant de plusieurs segments de mémoire pour la commande d'affichage d'une ligne de balayage, ce qui permet de manipuler les données enregistrées dans lesdits segments de mémoire pendant que les données bloquées  
25 sont extraites indépendamment dudit segment de mémoire.

14. Procédé selon la revendication 13, caractérisé en ce qu'il consiste à traiter en parallèle des données pour plusieurs lignes de balayage.

15. Procédé selon la revendication 12, caractérisé en ce que les données sont traitées en ordres s'étendant en longueur, englobant une ligne de balayage contenant les pixels à modifier (Y), un premier point à modifier (Xs) et un dernier point à modifier (Xe).

16. Procédé selon la revendication 12, caractérisé

en ce que des éléments de base graphiques sont transformés en ordres de remplissage de ligne horizontale transmis auxdits segments de mémoire.

5 17. Segment de mémoire à utiliser dans une  
mémoire rapide à architecture parallèle, caractérisé en  
ce qu'il comporte un réseau (50) de mémoire vive compre-  
nant des éléments d'enregistrement disposés en rangées  
et colonnes, une unité logique et arithmétique (56) qui,  
en réponse à des signaux de commande, enregistre des  
10 données dans le réseau de mémoire, accède à ces données  
et travaille sur elles, et des moyens (62) de commande  
destinés à diriger l'unité logique et arithmétique pour  
l'accès, le travail et l'enregistrement de données, un  
accès simultané à et un travail simultané sur tous les  
15 éléments de mémoire d'une rangée étant possibles.

18. Segment de mémoire selon la revendication 17,  
caractérisé en ce que ledit segment comprend un circuit  
intégré à semi-conducteur.

20 19. Segment de mémoire selon la revendication 17,  
caractérisé en ce que les moyens de commande et l'unité  
logique et arithmétique réagissent à des ordres de façon  
à travailler sur des sous-ensembles d'une rangée de  
mémoires sans modifier des parties non choisies, ledit  
sous-ensemble pouvant être modifié d'une opération à la  
25 suivante.

20. Segment de mémoire selon la revendication 17,  
caractérisé en ce que l'unité logique et arithmétique réagit  
en outre à des données de combinaison de trame pour accé-  
der à, travailler sur et enregistrer des données dans  
30 ledit réseau de mémoire vive.

21. Segment de mémoire selon la revendication 20,  
caractérisé en ce que l'unité logique et arithmétique réagit  
en outre à des ordres de façon à modifier des données  
enregistrées en exécutant des opérations de logique  
35 booléenne sur des parties spécifiées des données accédées

et sur la combinaison de trame.

22. Segment de mémoire selon la revendication 21, caractérisé en ce qu'il comporte en outre une bascule (58) destinée à recevoir des données des moyens de mémoire vive pour permettre une manipulation continue des données enregistrées dans ledit réseau de mémoire en même temps que les données placées dans la bascule sont extraites indépendamment dudit segment de mémoire.

23. Segment de mémoire selon la revendication 22, caractérisé en ce qu'il exécute des ordres comprenant quatre éléments: (i) la rangée du réseau de mémoire sur laquelle travaille ladite unité logique et arithmétique, (ii) les premier et dernier éléments de ladite partie de données accédées que l'unité logique et arithmétique doit modifier, (iii) la combinaison de bits de la combinaison de trame, (iv) l'opération de logique booléenne que l'unité logique et arithmétique doit exécuter.

24. Segment de mémoire selon la revendication 23, caractérisé en ce que les moyens de commande comprennent des moyens destinés à enregistrer un certain nombre desdites combinaisons et à présenter l'une d'elles à un générateur de combinaison à chaque fois que l'un des ordres s'étendant en longueur est présenté au segment de mémoire.

25. Segment de mémoire selon la revendication 22, caractérisé en ce que ladite combinaison peut être modifiée au moyen d'opérations connues de logique booléenne avant d'être utilisée par l'unité logique et arithmétique pour travailler sur ladite partie de données accédées.

26. Segment de mémoire selon la revendication 17, caractérisé en ce que les moyens de mémorisation de données enregistrent des données de pixels sous forme de valeurs d'intensité.

27. Segment de mémoire selon la revendication 17, caractérisé en ce que les moyens de mémorisation de

données comprennent plusieurs (K) positions d'enregistrement pour chaque valeur de l'intensité de pixels (K bit), lesdites positions d'enregistrement étant agencées de manière qu'un bit de chacun des pixels d'une ligne de balayage soit accessible en même temps pour tous les bits.

28. Segment de mémoire selon la revendication 27, caractérisé en ce que chaque rangée du réseau de mémoire contient l'un des grands K bits d'intensité d'une rangée de pixels.

29. Segment de mémoire selon l'une des revendications 27 et 28, caractérisé en ce qu'il comporte en outre une structure binaire arborescente pour produire des valeurs d'intensité de pixels interpolées.

30. Segment de mémoire selon la revendication 29, caractérisé en ce que les valeurs interpolées sont calculées en même temps pour tous les pixels choisis sur une ligne de balayage, lesdites valeurs étant produites pour tous les pixels choisis, un bit à la fois.

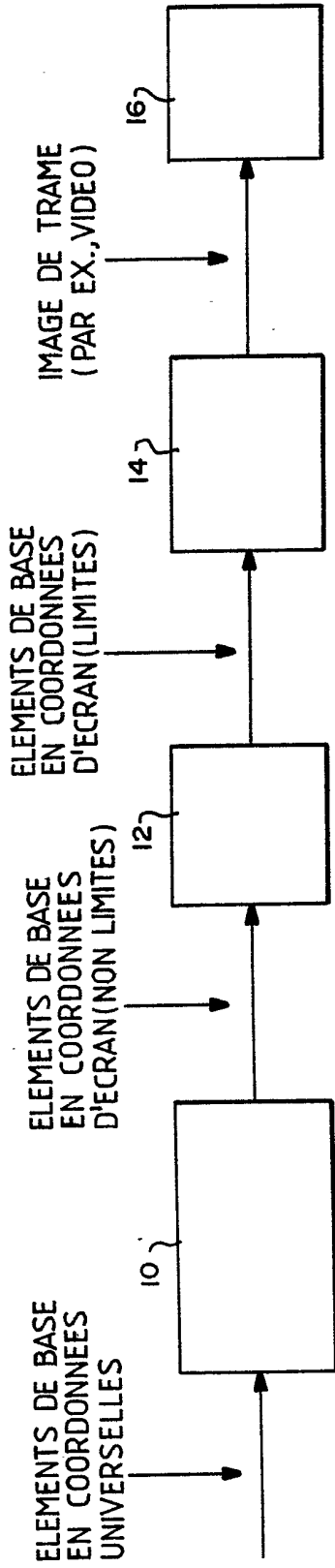


FIG. - 1

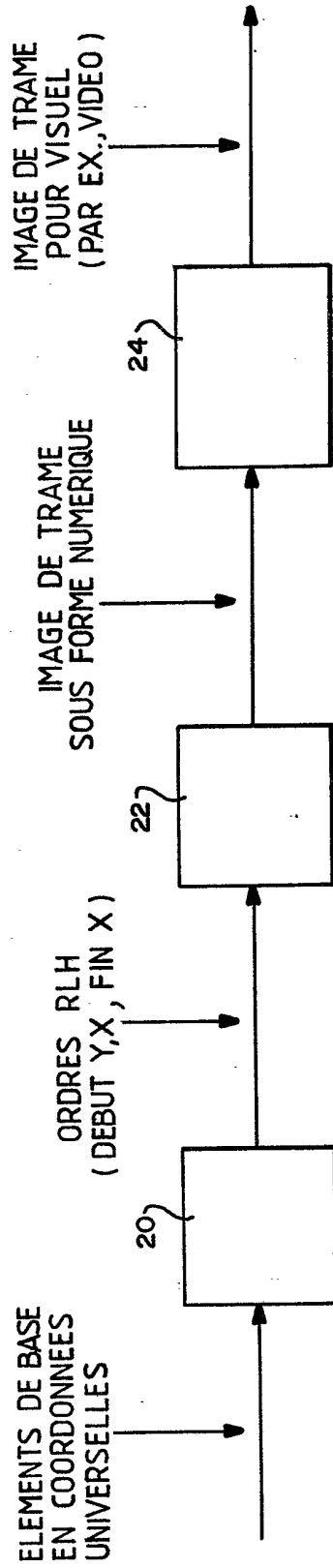


FIG. - 2

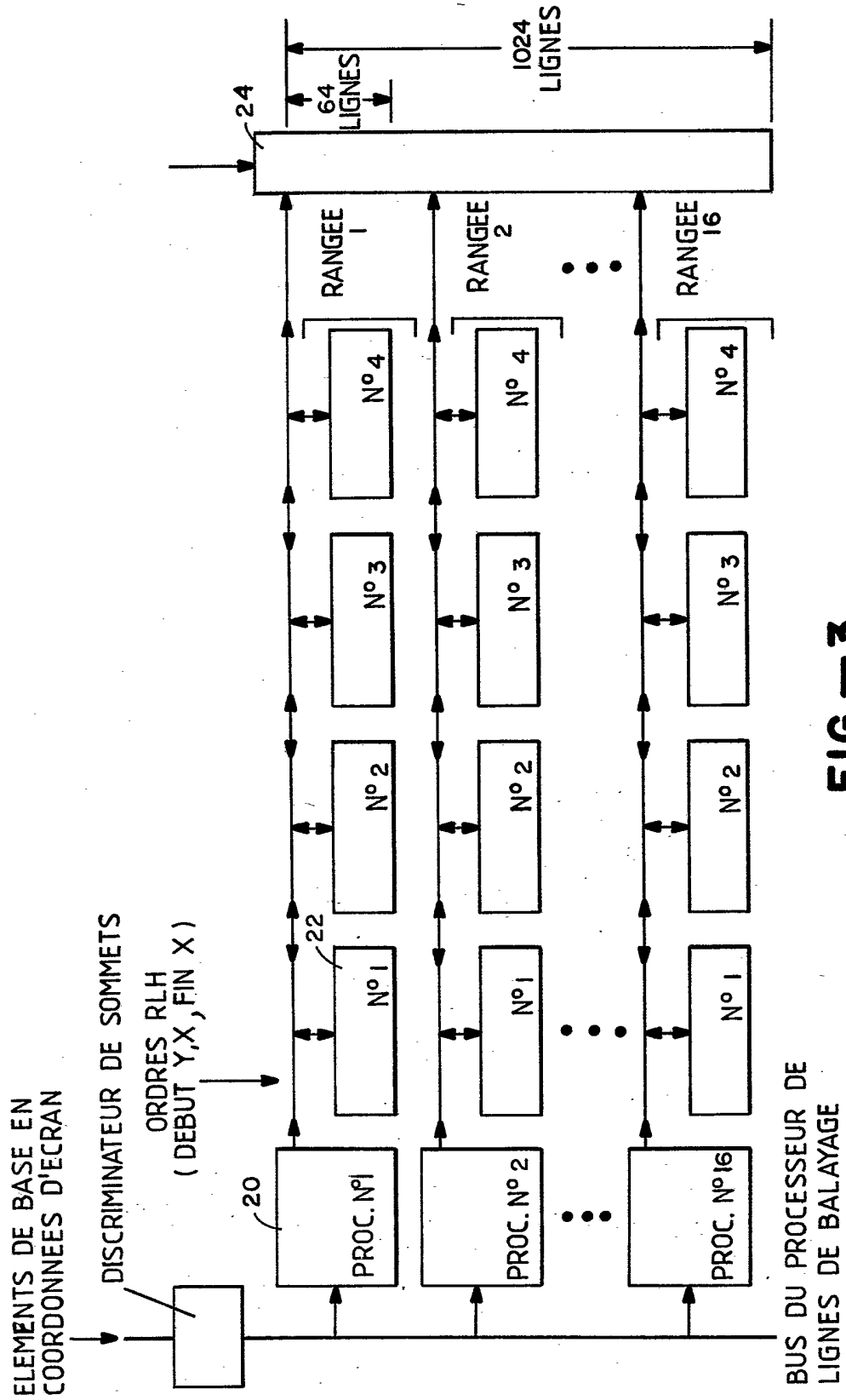


FIG.-3

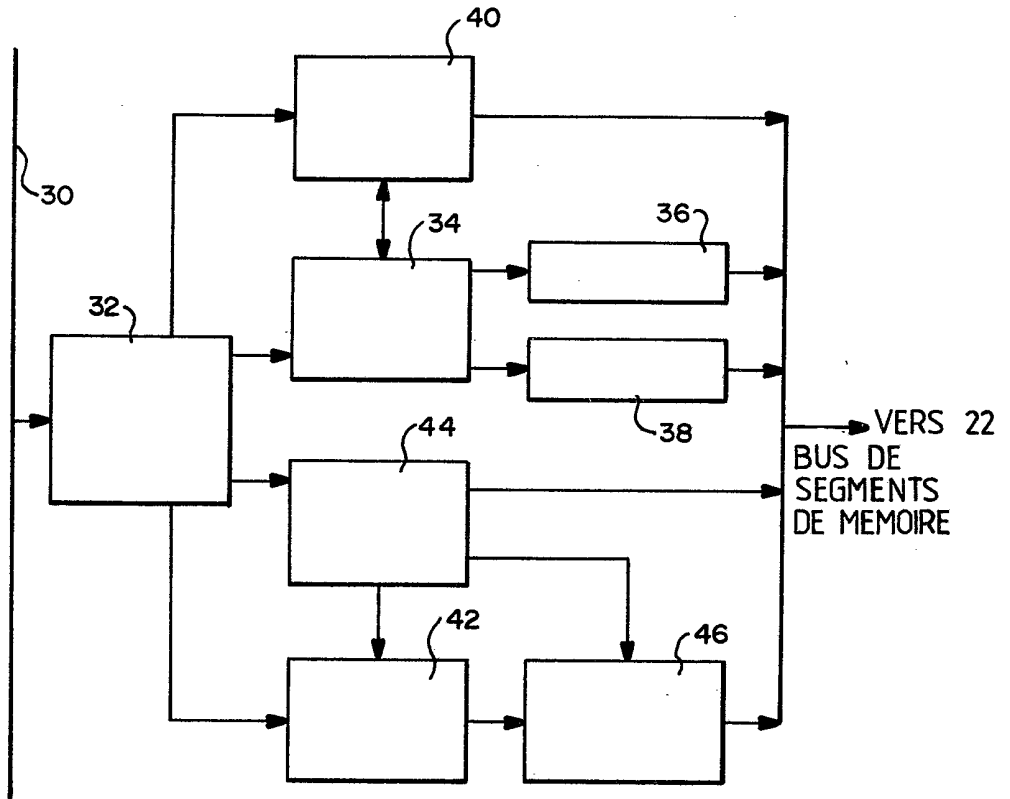


FIG. - 4

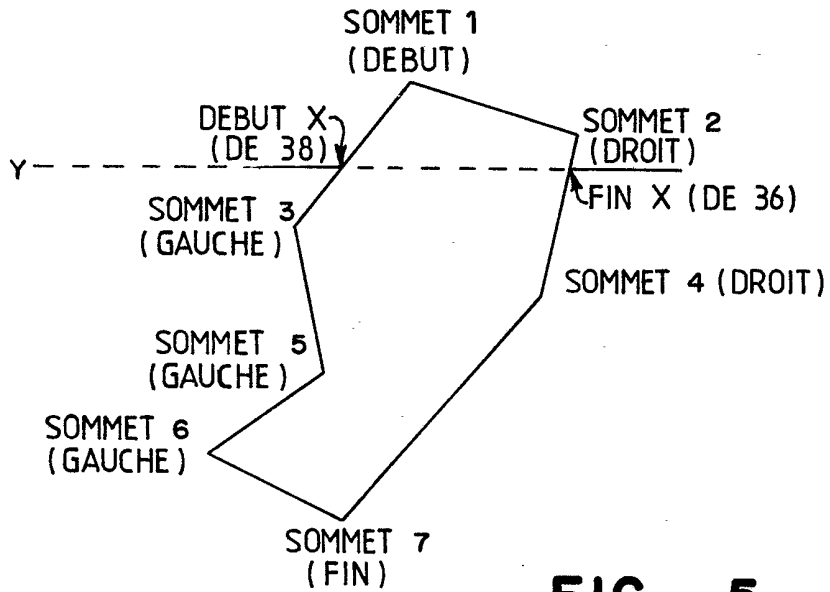


FIG. - 5

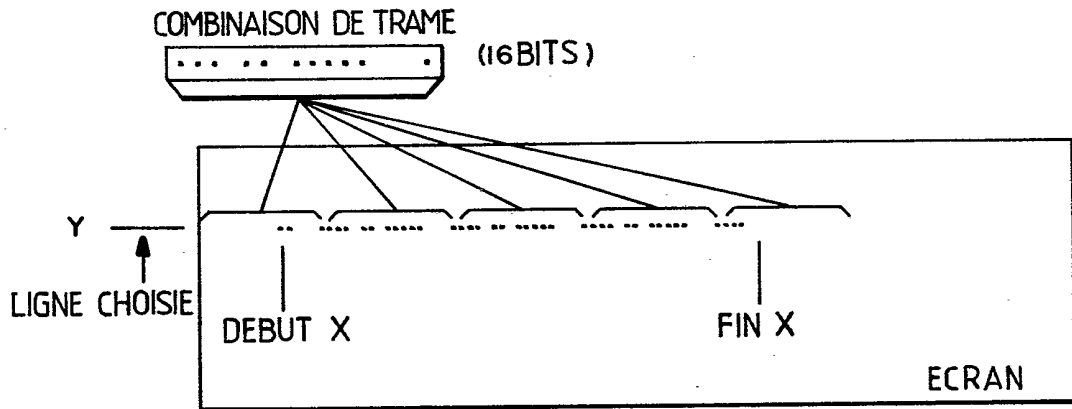


FIG. — 6

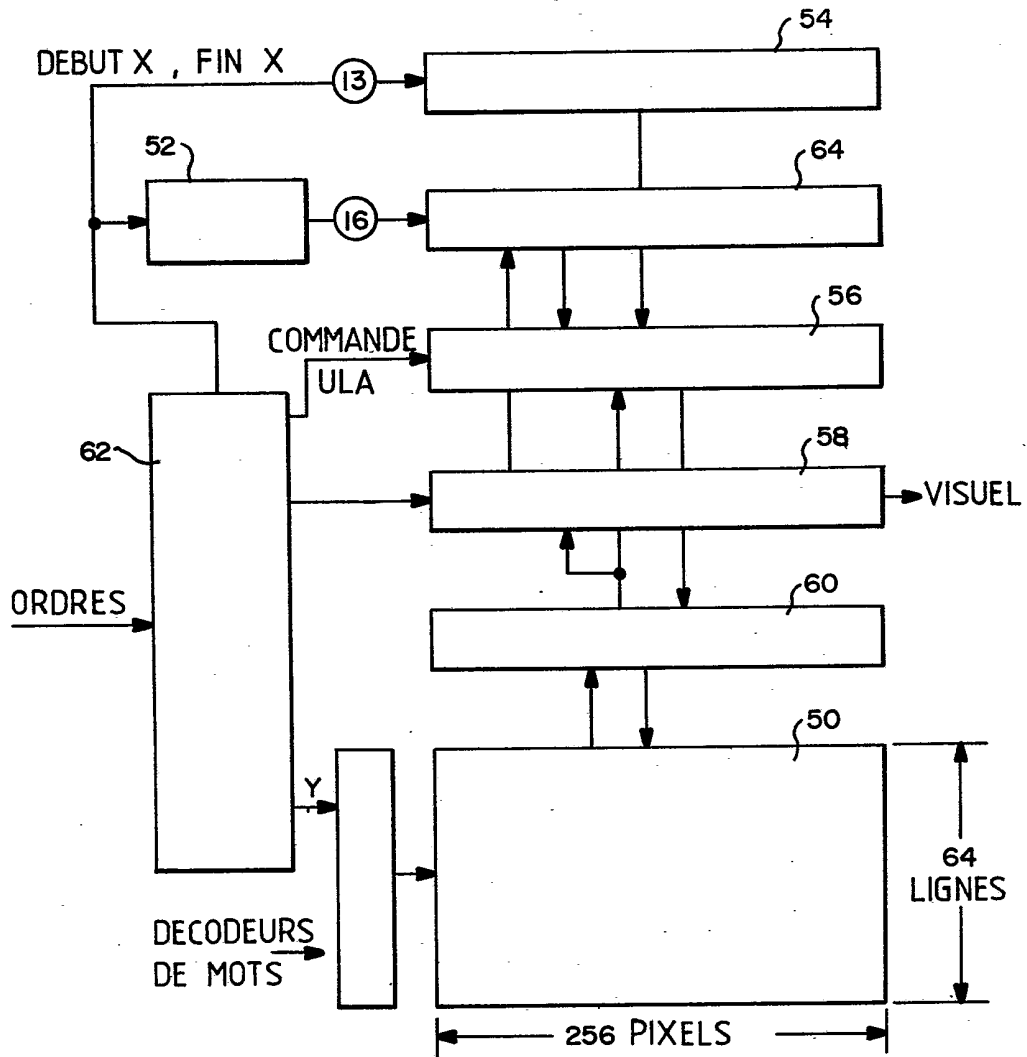
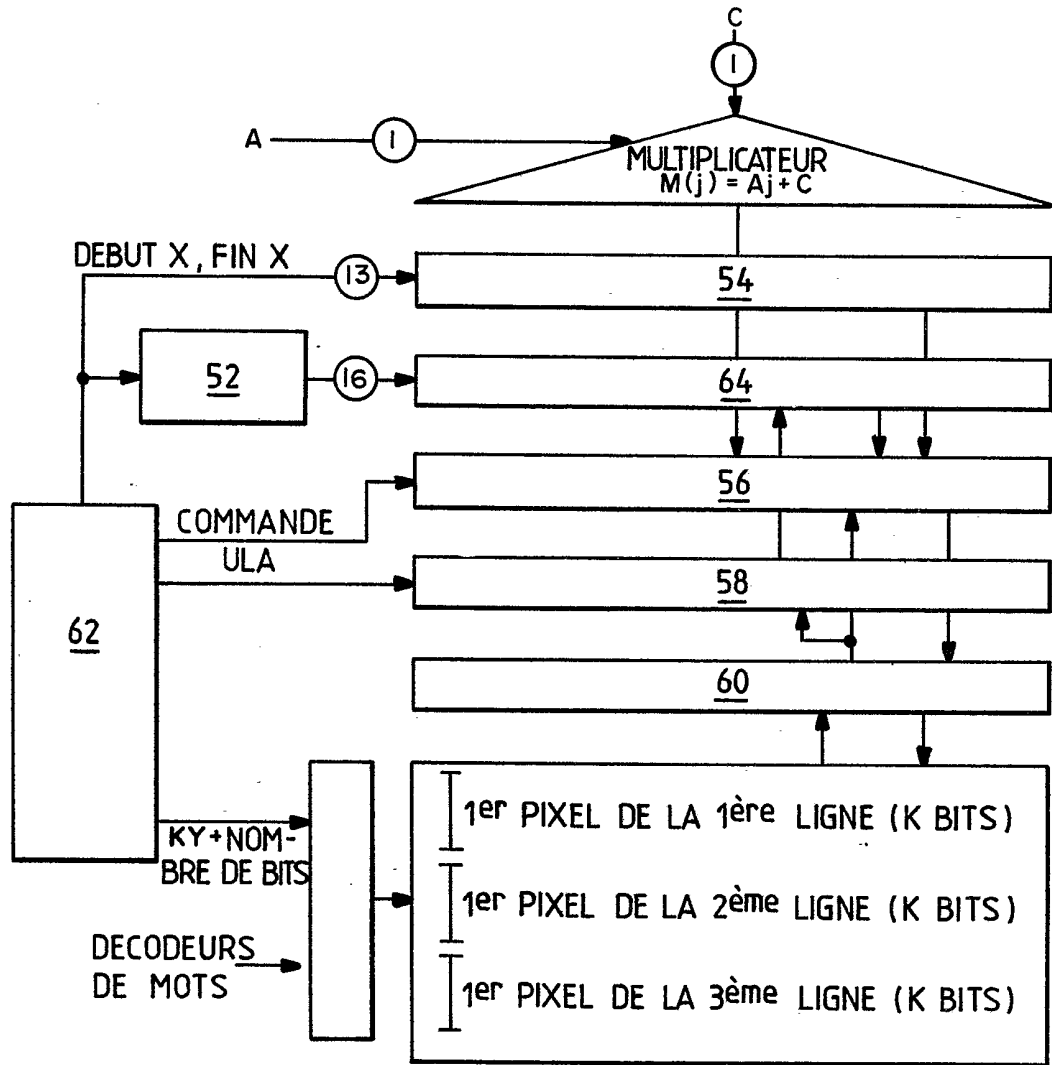
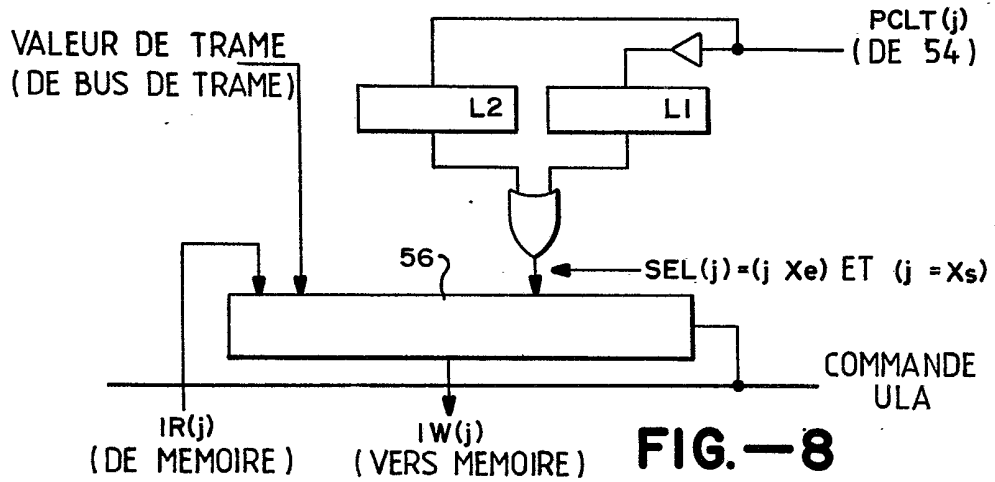


FIG. — 7



UNE COLONNE **FIG.—12**



**FIG.—8**

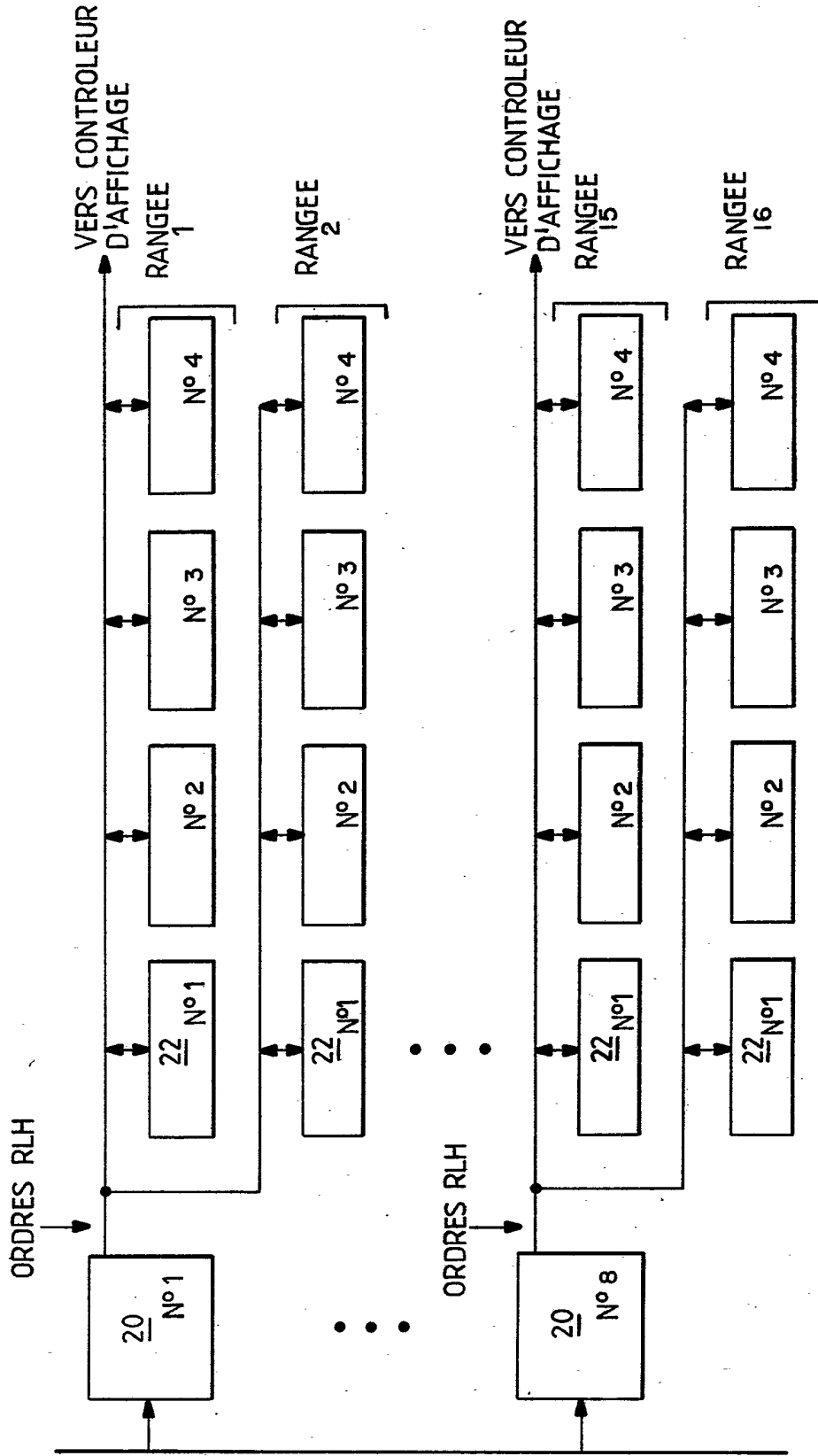


FIG.- 9

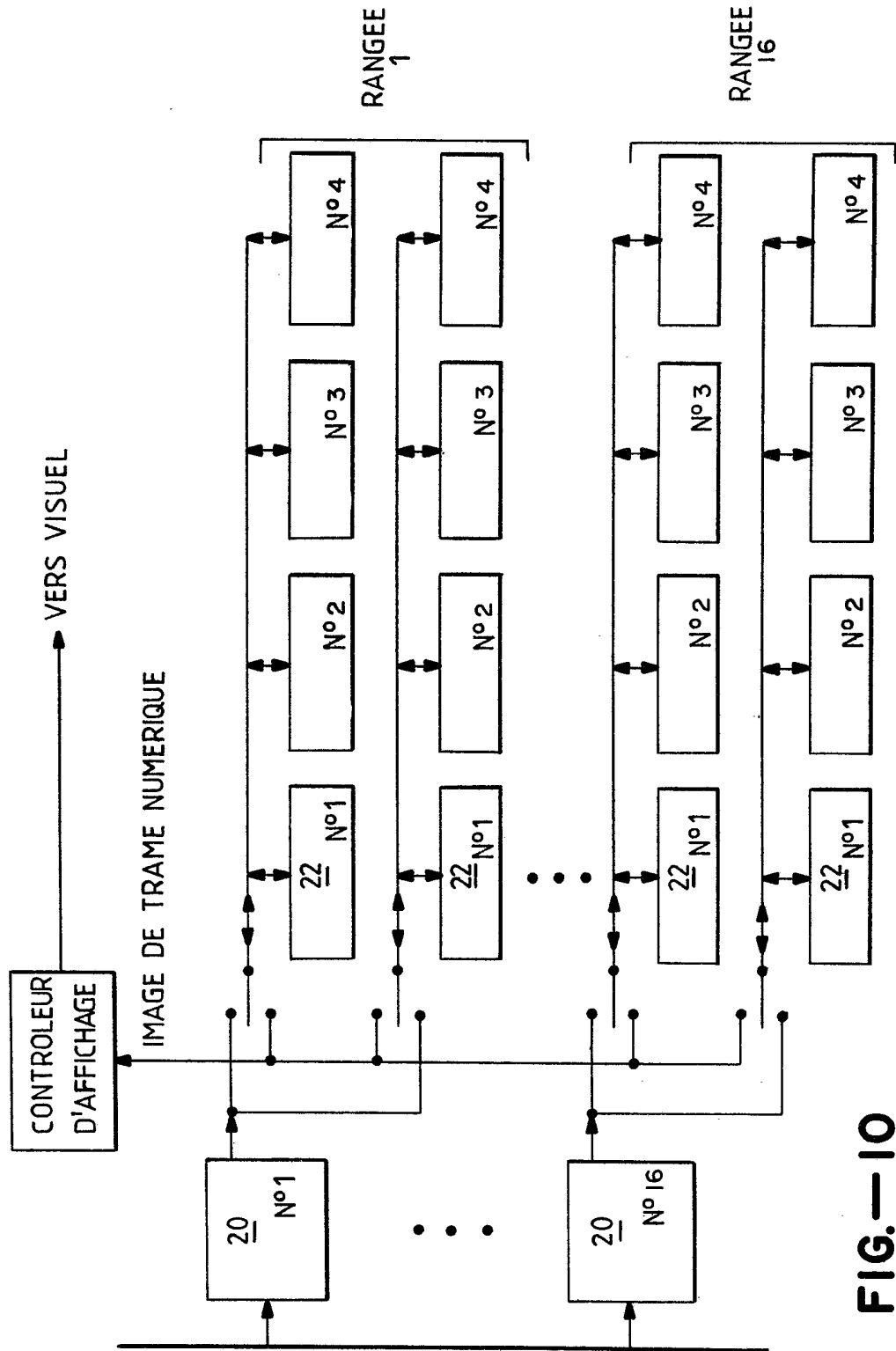


FIG.—10

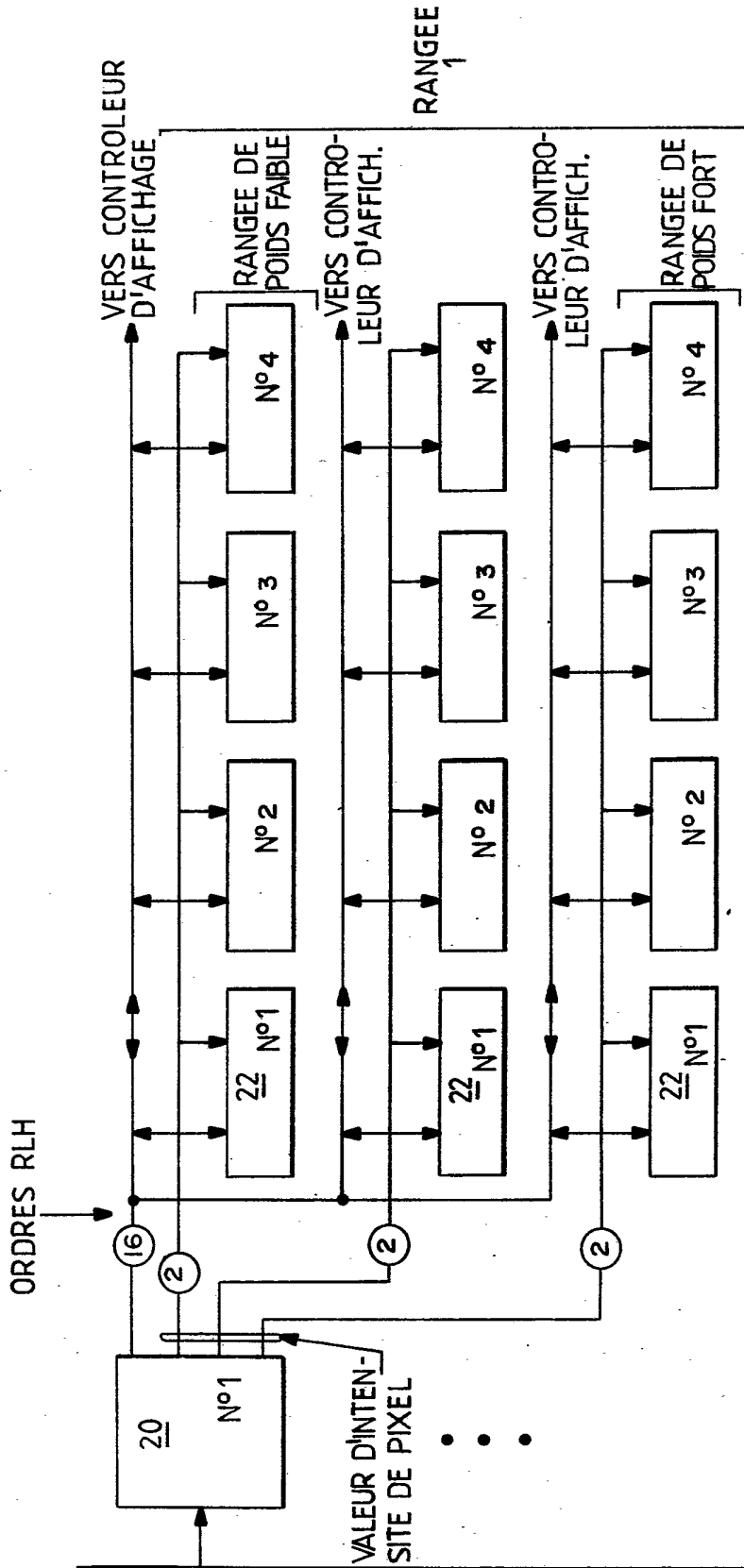
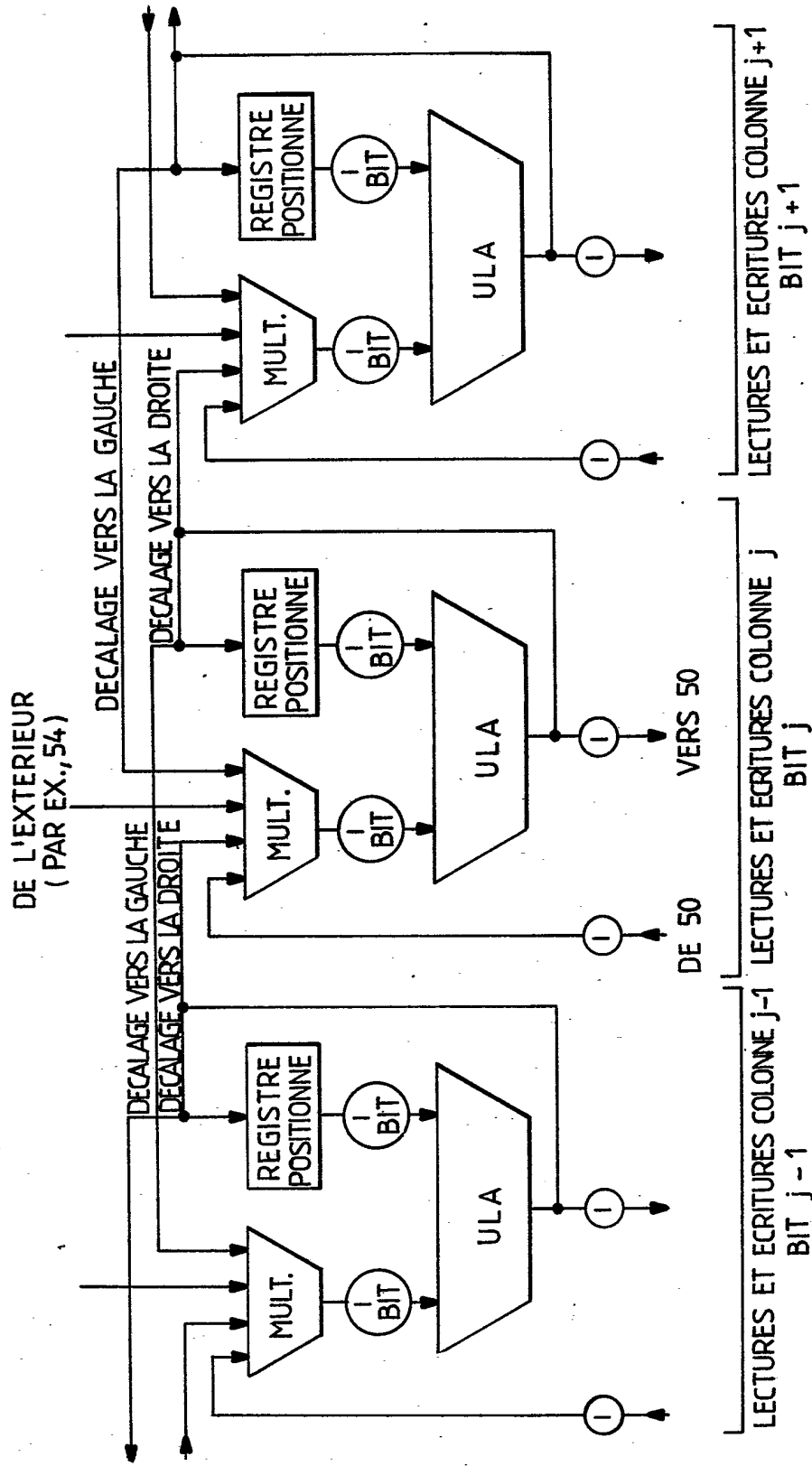


FIG. - II





**FIG. — 14**