(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 557 713**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **83 19467**

(51) Int Cl⁴ : G 06 F 15/20, 3/02, 3/16, 3/153; G 09 G 1/02.

(12) DEMANDE DE BREVET D'INVENTION A1

(22) Date de dépôt : 6 décembre 1983.

(30) Priorité :

(43) Date de la mise à disposition du public de la demande : BOPI « Brevets » n° 27 du 5 juillet 1985.

(60) Références à d'autres documents nationaux apparentés :

(71) Demandeur(s) : *HAURE Jean-Jacques.* — FR.

(72) Inventeur(s) : Jean-Jacques Haure.

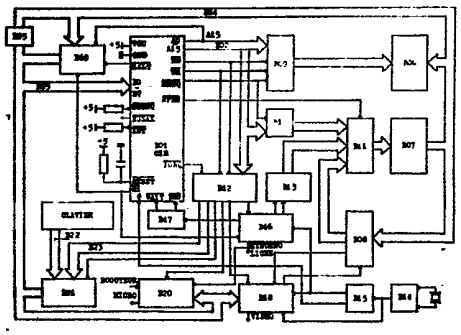
(73) Titulaire(s) :

(74) Mandataire(s) :

(54) Micro-ordinateur à usage domestique comportant des possibilités d'affichage vidéo, de lecture de clavier et de liaison à un magnétophone.

(57) Micro-ordinateur à usage domestique et d'enseignement comportant des possibilités d'affichage vidéo, de liaison à un magnétophone et de lecture d'un clavier à touches multi-fonctions produisant des caractères alpha-numériques, graphiques ou des fonctions du langage BASIC et d'accès au langage machine.

Il est constitué d'une unité centrale de traitement B 01 connectée à de la mémoire vive B 06 contenant des informations à sauvegarder sur un magnétophone via l'interface B 20 ou à afficher vers un écran pour lequel l'information vidéo est stockée dans un registre B 18 grâce à un code rangé dans un tampon B 08 afin d'adresser la mémoire morte B 07 via des multiplexeurs B 11. Les signaux de synchronisation sont générés par de la logique B 16. Le clavier est lu grâce à une interface B 21 et est décodé par la mémoire morte.



FR 2 557 713 - A1

Vente des fascicules à l'IMPRIMERIE NATIONALE, 27, rue de la Convention — 75732 PARIS CEDEX 15

La présente invention se rapporte au domaine des calculateurs numériques et concerne en particulier les micro-calculateurs ou micro-ordinateurs dotés de possibilités d'affichage vidéo, de lecture de clavier et de connexion avec des magnétophones.

Les quelques dernières années ont montré l'intérêt du public et de l'enseignement pour des calculateurs et micro-ordinateurs de faible prix mais ayant de bonnes performances. Pour pouvoir allier ces deux critères, il faut faire preuve d'ingéniosité afin de pouvoir utiliser des éléments faisant normalement partie de tout foyer susceptible d'utiliser un micro-ordinateur. Ainsi, le téléviseur peut faire office de terminal vidéo et le magnétophone peut servir de mémoire de masse permettant de stocker des informations afin de pouvoir les récupérer aisément. De plus, un tel système doit être suffisamment universel afin de pouvoir s'adapter aux multiples applications auxquelles les utilisateurs les destinent.

Un tel micro-ordinateur doit donc être capable de générer des signaux d'images compatibles avec les téléviseurs aux normes du pays concerné et doit pouvoir coder et décoder des informations afin de les mémoriser sur la plupart des magnétophones que l'on peut trouver dans le commerce avec lesquels il doit pouvoir se connecter grâce à des interfaces compatibles. De plus, pour permettre son utilisation immédiate, il doit être doté d'un langage de programmation répandu et facilement assimilable, par exemple le BASIC. Afin qu'il ne soit pas limité dans ses performances et pour le rendre attractif auprès de la clientèle visée, le langage de programmation propre au microprocesseur utilisé doit être facilement accessible. Ces deux langages devant être résidents.

Pour pouvoir allier ces performances aux critères de coût, il est nécessaire d'utiliser un microprocesseur peu coûteux et d'optimiser ses performances grâce à des montages électroniques astucieux. Ces montages pouvant être soit indépendants, soit commandés par des signaux générés par un logiciel spécifique associé au microprocesseur. Ce logiciel

devant obligatoirement être résident, donc implanté dans des unités de stockage de type mémoire morte.

Comme on le verra plus loin, le micro-ordinateur décrit ici convient de manière idéale aux applications grand public et à l'enseignement. Il offre une large gamme de possibilités non limitatives dont l'affichage de caractères alphanumériques et graphiques, le transcodage d'informations pour une connexion à un magnétophone à cassettes, un langage de programmation de type BASIC facilement accessible et, possibilité que l'on ne trouve pas dans les micro-ordinateurs de techniques antérieures destinés aux types d'applications décrites auparavant, l'accès facile au langage machine propre au microprocesseur utilisé. De plus ce micro-ordinateur est conversationnel en français et peut être adapté à d'autres langues.

La figure 1 représente le schéma synoptique de l'architecture matérielle complète du micro-ordinateur suivant l'invention décrite ci-après.

Le schéma de montage de la figure 2 représente les interconnexions et les moyens logiques associés afin de pouvoir utiliser la mémoire morte conformément à l'invention décrite.

Les figures 3 et 4 représentent le sous système d'affichage du micro-ordinateur suivant l'invention. La figure 3 est un schéma de montage représentant les circuits de génération du signal vidéo. La figure 4 est le schéma de montage des circuits destinés à produire le signal de synchro-ligne dont la forme d'onde est représentée en figure 5.

La figure 6 représente le principe de matrice d'affichage d'un caractère destiné à l'écran.

La figure 7 indique l'organisation d'un octet dont les bits sont utilisés pour la traduction d'une touche appuyée sur le clavier en un code représentatif d'un caractère ou d'une fonction.

Les figures 8a et 8b représentent la suite des instructions contrôlant le matériel afin de générer un affichage sur l'écran.

L'organisation de la matrice des touches du clavier, en QWERTY ou AZERTY, est présentée en figure 9. Ce clavier est scruté par la suite des instructions de la figure 10 afin de produire un code qui, après traitement par la suite des instructions des figures 12, 13a, 13b, 13c et grâce au tableau de la figure 14, permet de créer un nouveau code significatif de la touche appuyée conformément aux tableaux des figures 11a et 11b.

La figure 15 montre la procédure de sauvegarde des informations sur une cassette via l'interface de la figure 16. Cette sauvegarde se fait en produisant des signaux conformes aux figures 19 et 20 contrôlés par la suite des instructions des figures 17 et 18.

Bien que la description soit faite en faisant référence à un type déterminé de microprocesseur, il apparaîtra clairement au spécialiste de cette technique que les concepts inventifs décrits ici peuvent être utilisés avec d'autres microprocesseurs.

Bati autour d'un microprocesseur B01, aussi appelé CI8, de type Z80 et pour lequel tous les signaux sont figurés, le système représenté en figure 1 comprend un bus d'adresses B02 composé des lignes A0 à A15 issues de B01, d'un bus de données B03 qui, après passage dans le bloc bidirectionnel B05, prend le nom B04 afin d'être relié à de la mémoire de type RAM B06 et de type ROM B07 ainsi qu'à un registre tampon B08. La mémoire de type ROM stocke des programmes et des données dont une partie sera expliquée par la suite.

Le bus B02 et les signaux $\overline{RD}$, $\overline{WR}$, $\overline{MREQ}$ issus de B01 sont présentés à un décodeur B09 pour adresser la mémoire RAM B06, à un décodeur B10 pour générer des signaux qui sont présentés à un multiplexeur B11, et à un décodeur B12 qui, grâce à $\overline{IORQ}$ issu de B01, permet de sélectionner au choix les circuits de contrôle d'entrée-sortie B21 du clavier, de l'interface cassette B20, et des interfaces de génération des signaux de synchro B16 et vidéo B18 destinés à un écran à tube cathodique.

La mémoire morte B07 est adressée par un multiplexeur B11 dont les entrées proviennent soit de B10 ou soit d'un compteur par huit B13 et du registre tampon B08 selon la valeur du signal $\overline{RFSH}$ issu de B01.

Un oscillateur B14, qui peut être contrôlé par un quartz, produit des signaux à une fréquence de l'ordre de 6,5 Mhz destinés à la génération du signal vidéo et à un diviseur par deux, B15, afin de générer l'horloge ϕ de B01 et de piloter la génération des signaux de vidéo et de synchro-ligne grâce à B16 qui, lui-même, contrôle le compteur B13, le circuit B17 de génération des signaux $\overline{NMI}$ et $\overline{WAIT}$ ainsi que le bloc B20 pour produire un signal vers la cassette.

Le signal vidéo, issu de B18, est généré grâce à une information présente sur B03, quand les portes à collecteur ouvert symbolisées par B19 le permettent, et à un bit issu du tampon B08, contrôlé lui-même par ce bloc de génération de vidéo.

Les signaux MICRO destinés à la cassette et ECOUTEUR provenant d'un magnétophone sont contrôlés par B20 qui utilise le signal de synchro-ligne et le bus B03.

Enfin, le clavier est décodé par B21 grâce à un bus B22 et au bus B23 provenant de B12.

La figure 2 montre les différentes manières dont les mémoires mortes B07 sont adressées. Dans le but de prévoir l'avancement de la technologie et de faciliter l'industrialisation, le décodage est fait de sorte qu'il soit possible de connecter deux mémoires ROM ou REPROM de 8 Kilo-octets chacune (la première, CI5, à partir de l'adresse 0000H et la seconde, CI6, à partir de l'adresse 2000H) ou bien une seule mémoire morte CI5 de 16 Kilo-octets. La sélection des types de mémoires choisis se fait à l'aide de connexions qu'il faut établir entre des points bien définis. Ainsi, il faut relier a à c si deux mémoires de 8 Kilo-octets sont utilisées ou bien a à b si le choix d'une seule mémoire de 16 Kilo-octets a été fait. Si CI5 est du type ROM 8 Kilo-octets, il faut établir les liaisons m-p, d-e, g-h. Si cette

mémoire est du type REPROM de même capacité, il faut réunir m-n, d-f, g-i, j-l. Enfin, dans le cas d'une seule mémoire de 16 Kilo-octets, CI6 n'étant alors pas utilisée, il faut relier : m-n, d-f, g-i, j-k. Le choix d'une ROM de 8 Kilo-octets pour CI6 implique les liaisons u-w, q-r ; le cas d'une REPROM étant sélectionné par les liaisons u-v, q-t.

Cette figure montre aussi clairement que les adresses fournies aux mémoires CI5 et CI6 peuvent avoir deux origines possibles multiplexées par les circuits CI1, CI2 et CI3 correspondant au bloc B11. Lorsque le signal $\overline{RFSH}$, généré par le microprocesseur, est au niveau logique 1, l'adresse est fournie directement par ledit microprocesseur : les lignes A0 à A9 sont sélectionnées par les trois multiplexeurs, les lignes A10 à A12 sont présentées directement aux mémoires et il faut que les portes logiques 201, 202 et 203 vérifient l'équation $A15+A14+\overline{MREQ}=0$ afin d'imposer un niveau bas sur certaines entrées de CI5 via CI1. Dans ce cas, lorsque la ligne A13 est au niveau bas, la mémoire CI5 est sélectionnée alors que les portes 204 et 205 imposent un niveau haut sur deux entrées de CI6 pour ne pas sélectionner ce circuit. Quand A13=1, seule la mémoire CI6 est choisie.

Quand $\overline{RFSH}=0$, ce qui ne se produit que pendant les deux dernières périodes d'horloge du microprocesseur lors de la recherche d'une instruction, les trois multiplexeurs sélectionnent douze autres lignes pour adresser les mémoires CI5 et CI6. Il sera vu plus loin que cet adressage servira à l'acquisition d'une information nécessaire à la génération du signal vidéo. Simultanément, le contenu du registre I du microprocesseur, chargé par programme par la valeur prédéfinie 1CH, est présenté sur les lignes A15 à A8 du bus B02. Les entrées A1 et A2 de CI1 étant respectivement aux niveaux logiques 0 et 1, seule la mémoire CI5 est sélectionnée et est adressée par A13=0, A12=1, A11=1, A10=1. Les signaux arrivant aux entrées A3 à A9 de CI5 sont issus du circuit CI4, bloc B08, dans lequel a été stocké auparavant le code du caractère à afficher selon un processus qui sera expliqué par la suite. Enfin, les entrées A0 à A2 de CI5 sont reliées à un compteur

par huit, CI7 (bloc B13), qui peut être initialisé par SRO et incrémenté par $\overline{SYNC}$, inverse du signal SYNCHRO-LIGNE, afin de faire varier l'adresse de la mémoire dans le but de pointer consécutivement les huit lignes composant la matrice d'affichage du caractère dont le code est mémorisé dans CI4. Le changement du choix d'une ligne parmi huit est donc fait à chaque ligne d'écran nouvelle demandée par le signal SYNCHRO-LIGNE.

Ce procédé permet d'adresser une zone mémoire comprise entre les bornes 1C00H et 1FFFH incluses où se situent l'ensemble des 128 matrices d'affichage des caractères. Pour un code de caractère donné, l'adresse du début de sa matrice se trouve par la formule : 1C00H+(8x code). La prise des huit lignes de la matrice se fait grâce au compteur CI7 incrémenté par le signal $\overline{SYNC}$ à chaque nouvelle ligne affichée sur l'écran. Par exemple, la matrice du caractère "A", montrée en figure 6 et dont le code est 41H car codé en ASCII, se situe des adresses 1E08H à 1E0FH. Chaque bit des huit octets indique si le point doit être allumé ou éteint sur l'écran.

Les circuits servant à générer le signal vidéo sont représentés en figure 3 où seuls les signaux nécessaires à l'explication sont représentés.

Lors de cette génération, il sera vu plus loin que le bit A15 du bus d'adresses est imposé à l'état haut et les bits A14 à A0 contiennent l'adresse du code du caractère à afficher.

Le microprocesseur CI8 rentre dans le cycle de recherche d'instruction pendant lequel $\overline{M1}$=0 durant les deux premières périodes de l'horloge qui pilote le microprocesseur. Dans le cas où le code du caractère est différent de 1111 1111 en binaire, soit FFH en hexadécimal, l'équation

$$\overline{((\overline{F0}.\overline{F1}.\overline{F2}.\overline{F3}.\overline{F4}.\overline{F5}.\overline{F6}.\overline{F7}) . \overline{HALT}) . A15 + \overline{M1}}$$

, vérifiée par les portes logiques 301 à 304, donne la valeur logique 1 qui, présentée aux huit inverseurs à collecteur ouvert CI20 à CI27, force les huit bits D0 à D7 du bus de données du microprocesseur à l'état bas grâce aux résistances de rappel RR2 (bloc B05).

Ainsi, le microprocesseur lira 00 comme code d'instruction, c'est à dire l'instruction NOP qu'il exécutera et qui dure quatre périodes d'horloge. Dans la suite, une période d'horloge sera appelée T. Le fichier d'affichage se trouve en mémoire vive pour laquelle le bit A15 n'est pas décodé. Ainsi les bits A0 à A14 adresses réellement le code du caractère à afficher qui se trouve alors présenté sur le bus F0 à F7, B04, et qui est mémorisé dans le registre CI4 (figure 2) en tout début de la troisième période d'horloge T grâce à SLE=MREQ.$\overline{RFSH}$.$\overline{\Phi}$=0 comme le vérifient les portes 305 et 306. Pendant les deux dernières périodes d'horloge des quatres nécessaires à l'exécution de cette instruction NOP, le signal $\overline{RFSH}$ est au niveau bas et permet ainsi d'adresser la ROM (figure 2) avec l'adresse de la matrice du caractère. L'octet contenu à cette adresse, représentant une ligne sur les huit affichables, est alors lu et présenté sur les lignes F0 à F7 ainsi que sur les entrées D0 à D7 de CI8 via les résistances RR2 et P0 à P7 de CI9 car, lors de ces deux dernières périodes d'horloge, $\overline{M1}$ est imposé au niveau haut enlevant ainsi l'inhibition qui forçait des 0 sur le bus B03.

Lors de la dernière demi-période du cycle d'exécution du NOP, le signal $\overline{MREQ}$ est au niveau haut. Le décodage par les portes 307 et 308 de la sortie Q du tampon CI13, mémorisée à 0 par SLE auparavant, $\overline{MREQ}$ et une impulsion brève donnée par CI12 permet de mémoriser, par SL=0, dans le registre à décalage CI9 l'octet représentant une ligne à afficher. Simultanément et grâce aux portes 309, 310 et 313, le bit de poids fort du code, CI407 provenant de CI4, est mémorisé par la bascule RS constituée des portes 311 et 312. Ce signal sera utilisé pour générer une inversion éventuelle du signal vidéo pour ce caractère.

Les huit impulsions de décalage du registre CI9 sont générées par CI10 à la fréquence de l'ordre de 6,5 Mhz lors de l'exécution de l'instruction suivante dont les quatre premiers cycles d'horloge du microprocesseur, à 3,25 Mhz car divisée par deux par CI11 (bloc B15), servent à la recherche de l'instruction. Les impulsions de décalages sont générées grâce à la bascule CI10 de type D dont la sortie $\overline{Q}$

est reliée à l'entrée CLR, active au niveau bas. Ainsi, chaque front montant présenté sur l'entrée CK provoque le passage de la sortie Q à 1, car D est constamment à 1, et de $\overline{Q}$ à 0 qui, présenté sur l'entrée CLR, force les sorties Q et $\overline{Q}$ respectivement à 0 et 1. La durée brève de l'impulsion produite sur la sortie Q est très étroitement liée aux temps de propagations internes de la bascule utilisée.

Tant que le bit A15 est au niveau haut, le système est fait de manière telle qu'il fonctionne comme décrit précédemment pour générer l'affichage en simulant la lecture d'un NOP sauf si le code réel lu et présenté sur F0-F7 est FFH. Dans ce dernier cas, détecté grâce à la porte à huit entrées 301, les niveaux bas ne sont plus imposés sur le bus D0-D7 et le code FFH est réellement lu par le microprocesseur qui exécutera ainsi l'instruction correspondante, soit RST38.

Le signal vidéo issu de la sortie Q de CI9 peut être inversé grâce au OU EXCLUSIF 314 pour un caractère quelconque par le bit CI407, bit de poids fort du code de l'octet à afficher, ou bien à l'aide du OU EXCLUSIF 315 pour l'ensemble de l'écran par la sortie Q de la bascule CI14, initialisée à 1 par le $\overline{\text{RESET}}$ lors de la charge du condensateur C2 à travers la résistance R21, pouvant être modifiée pour chaque instruction provoquant $\overline{(\overline{\text{IORQ}}+\overline{\text{WR}})+(A3+\overline{A0}.A1.\overline{A2})} = 1$ et vérifiée par les portes 316 à 319. Ce qui se produit, par exemple, à l'exécution de l'instruction OUT (QF7),A.

Le schéma de la figure 4 est le détail du bloc B16 et montre le fonctionnement de la génération du signal de SYNCHRO-LIGNE destiné à l'écran. Ce signal peut être généré par le matériel seul, SLM, ou bien être contrôlé par logiciel pour prendre le nom SLL ; le choix étant fait par multiplexage.

Le signal SLM est construit à partir du signal inverse de l'horloge, de période T, du microprocesseur qui est divisée plusieurs fois par deux grâce aux compteurs binaires CI15 et CI16. Les 15ème et 208ème impulsions sont détectées par des portes afin de générer un signal durant 15 périodes d'horloge au niveau bas et 193 au niveau haut conforme à la

figure 5. La 15ème impulsion, détectée par les portes 401 et 403, fait passer la sortie Q de CI17 au niveau haut et la 208ème, détectée par 402, réinitialise les compteurs et la bascule, grâce à l'inverseur 404, afin de générer un nouveau signal identique. Pour éviter tout aléa de fonctionnement, quatre inverseurs, 405 à 408, sont reliés à la sortie Q de CI17 afin de produire un retard dans le signal présenté sur la porte OU 409.

Le signal SLL est commandé par le logiciel dont les instructions sont indiquées en figure 8a. L'ensemble de ce programme sera expliqué par la suite.

Au commencement de l'exécution de ce programme, ayant pour origine l'adresse 0038H, les sorties $\overline{Q}$ de CI18, Q de CI19 et CI20 sont au niveau haut. Lors de l'exécution de l'instruction OUT (OFF),A la sortie $\overline{IORQ}$ passe au niveau bas après la huitième impulsion d'horloge et, étant connectée à l'entrée PRESET de CI18 active au niveau bas, provoque $\overline{Q}$=0. Le signal $\overline{M1}$ passe au niveau bas lors de la recherche de l'instruction DEC C et, après inversion par 410, force Q de CI19 au niveau bas. Le front descendant de $\overline{M1}$ lors de la recherche de l'instruction JP NZ,INTER1 force la sortie Q de CI20, c'est à dire SLL, au niveau bas réinitialisant ainsi $\overline{Q}$ de CI18=1 car SLL est connecté à l'entrée CLR de CI18 active au niveau bas.

Que la condition testée lors de l'instruction JP NZ,INTER1 soit vérifiée ou non la durée de cette dernière est de 10T et une instruction qui dure 5T, RET Z ou RET NZ, est toujours exécutée ensuite pendant laquelle la sortie Q de CI19 repasse à 1 car l'entrée CK de cette bascule est excitée par $\overline{\overline{M1}}$. Lors de l'instruction suivante, le front descendant de $\overline{M1}$ fait repasser SLL au niveau haut.

Ainsi, le signal SLL est au niveau haut pendant les deux premières instructions qui durent en tout 15T puis SLL=0 pendant 15T. Le signal reste au niveau haut pendant la suite du programme qui,quelque soit la suite d'instructions exécutées, dure 31T car des instructions ne visant qu'à créer des retards sont exécutées dans la branche INTER1. Il sera vu par la suite que le JP (HL) se fait avec le bit

de poids fort de H égal à 1 pour forcer A15=1 afin de pouvoir générer le signal vidéo. Il a été vu que, dans ce cas, le microprocesseur exécute des instructions NOP, de durée 4T, pendant l'affichage des 34 colonnes qui constituent une ligne d'écran. Il rencontre ensuite l'instruction RST38, signifiée par FFH et de durée 11T. Cette dernière instruction fait ré-exécuter le programme de la figure 8a à partir de l'adresse 0038H. Le temps écoulé entre deux exécutions est donc de 15T+15T+31T+(4T x 34)+11T, soit 208T.

Ainsi la forme et la durée des signaux SLL et SLM sont identiques et conformes à la figure 5. Il est à noter que, l'horloge du microprocesseur étant de l'ordre de 3,25 Mhz, les durées de 208T et 15T correspondent respectivement à des temps de 64 et 4,6 micro-secondes donc compatibles avec les types de signaux de synchro-ligne qu'un écran à tube cathodique peut recevoir.

Le multiplexage entre SLL et SLM est assuré par les deux ET 411 et 412 ainsi que le OU 413. La commande de ce multiplexeur est faite par la bascule RS constituée de 414 et 415, elle-même commandée par la logique qui détecte les instructions OUT (OFE),A et OUT (OFD),A par la vérification de l'équation $\overline{WR}+\overline{IORQ}=0$ grâce à 416 combinée avec les bits A0 à A3 du bus d'adresses reconnus par les portes 417 à 426.

Quand l'instruction OUT (OFE),A est détectée, le signal SLM est présenté sur l'entrée CK de CI21 et le signal d'interruption $\overline{NMI}$ est généré vers CI8 à travers la porte OU 427. Quand l'instruction OUT (OFD),A est reconnue, le signal SLL est sélectionné et la sortie $\overline{Q}$ de CI21 est imposée au niveau haut afin d'inhiber par le OU 432 tout signal présent sur l'entrée CK de CI21 jusqu'au prochain front montant donné par SLL, ceci pour éviter tout aléa de fonctionnement. Le signal $\overline{SYNC}$ ainsi produit sert à incrémenter le compteur de lignes CI7 (figure 2)et, après inversion par la porte 433, fournit le signal SYNCHRO-LIGNE pour l'écran. La remise à zéro du compteur CI7 est assurée par SRO=1 quand l'équation $\overline{IORQ}+\overline{RD}+A0=0$ est vérifiée par les portes 416, 428, 429 et la bascule RS constituée des portes 430 et 431. Ce dernier cas se produisant par exemple lors de l'exécution de l'instruction IN A,(OFE).

Il sera vu par la suite que la parfaite synchronisation nécessaire entre les signaux SLM et SLL est assurée par logiciel.

Le fichier contenant les codes des caractères à afficher se trouve en mémoire vive entre les adresses 4030H et 43BDH incluses. Le contenu de cette zone de 910 octets se décompose en 26 lignes de 35 caractères codés pour la plupart en ASCII. Le dernier caractère de chaque ligne est toujours le code FFH correspondant à l'instruction RST38 du microprocesseur Z80.

Le principe d'affichage du caractère selon son code a été vu précédemment.

La figure 8a donne la liste des instructions du programme, commençant à l'adresse 004AH, gérant la partie utile de l'affichage. Le registre HL est initialisé par l'adresse de la première case du fichier d'affichage, soit 4030H. Le bit 7 de H est positionné à 1 afin d'imposer A15=1 lorsque l'instruction JP (HL) de l'adresse 0044H, qui provoquera ainsi PC=C030H, sera exécutée. Le registre B est initialisé par le nombre de lignes à afficher (1AH=26) et le registre C par 8, représentant le nombre de lignes par matrice de caractère. Le signal de SYNCHRO de la ligne à afficher est envoyé par les instructions des lignes 0053H à 0057H qui génèrent une forme d'onde représentée en figure 5 car les temps d'exécution de ces instructions correspondent à ceux des lignes d'instructions du programme générant les signaux de SYNCHRO-LIGNE expliqués auparavant; les retards générés des lignes 0058H à 005AH servent à conserver une durée de 31T jusqu'au JP(HL) inclus. Cette dernière instruction, avec HL=C030H, provoque A15=1 et les bits A14 à A0 pointent l'adresse du premier code à afficher. L'exécution du code situé en 4030H provoque alors la mise à 0 des huit bits du bus de données du microprocesseur afin de simuler un NOP pour pouvoir générer l'affichage dont l'explication détaillée a été faite auparavant. Le compteur programme du microprocesseur s'incrémente automatiquement à chaque instruction NOP lue et pointe ainsi successivement les 34

codes des caractères affichables de la ligne. Lorsque le code FFH est lu, ce qui arrive après le 34ème code affichable, il est reconnu par la porte 301 et est interprêté comme étant le code de l'instruction RST38 qui provoque alors la mise automatique en pile de l'adresse suivant ce code et l'exécution du programme de gestion des interruptions logicielles à l'adresse 0038H.

Celui-ci commence par envoyer un nouveau signal de synchronisation pour la nouvelle ligne et décompte le nombre de lignes affichées sur les huit que comporte la matrice du caractère grâce à l'instruction DEC C. Tant que ces huit lignes ne sont pas affichées, le saut en INTER1 permet, par l'instruction POP DE, d'annuler ce qui a été rangé en pile sans modifier HL qui tient toujours l'adresse du début de la ligne en cours d'affichage. Après un retard généré par les lignes 0047H et 0048H afin de conserver toujours la durée de 208T entre deux signaux de synchronisation, un nouveau JP (HL) fait afficher la ligne suivante de la matrice car le signal de SYNCHRO envoyé a incrémenté le compteur CI7 (figure 2) ... et ainsi de suite jusqu'à ce que les huit lignes de la matrice soient affichées, ce qui est détecté par C=0 à l'adresse 003BH. Dans ce cas, l'instruction POP HL a pour effet d'annuler l'adresse qui avait été sauvegardée en pile en réinitialisant HL par l'adresse du premier caractère de la ligne suivante éventuelle à afficher, décomptée par B. Le compteur de lignes pour la matrice est réinitialisé à huit par l'instruction SET 3,C avant le JP (HL) qui continue l'affichage.

Lorsque les 26 lignes sont affichées, ce qui se détecte par B=0, l'instruction RET Z située à l'adresse 0041^H termine le programme d'affichage et fait continuer l'exécution à l'adresse 0088H, c'est à dire après le CALL DISPLA (fig. 8b).

L'écran sur lequel l'affichage est produit est divisé en trois parties distinctes définies par le logiciel du micro-ordinateur.

Dans le cas de l'utilisation d'un écran de 625 lignes, où seulement 312 sont utilisées, le logiciel produit un signal de synchro trame qui dure 2 lignes suivi de 63 lignes

pour lesquelles la SYNCHRO -LIGNE est générée par le matériel grâce au signal SLM (figure 4). Ensuite les 26 lignes de caractères, composés chacun d'une matrice 8 x 8, sont affichées grâce au logiciel DISPLA (209 signaux de SYNCHRO-LIGNE sont alors produits par SLL) et enfin, les 38 lignes restantes sont produites pendant lesquelles SLM génére la SYNCHRO-LIGNE. Un nouveau cycle recommence ensuite pour rafraîchir l'image de l'écran. Ce rafraîchissement est fait toutes les 20 ms.

Un indicateur, représenté par le bit 3 d'une variable nommée ECRIMP, permet de savoir si le programme doit gérer la partie, sans affichage sur l'écran, haute ou basse afin de déterminer s'il faudra produire un affichage ou bien un signal de synchro-trame ensuite. Un registre A' contient en permanence le nombre de lignes sans affichage restant à produire. A l'initialisation du système, le bit 3 de ECRIMP est positionné de sorte de simuler la fin de la génération de la partie basse de l'écran et le registre A' est prédéfini à 1 afin de pouvoir produire le signal de synchro-trame dès que la première interruption $\overline{NMI}$, qui a été autorisée auparavant, interviendra et fera exécuter le programme dont les instructions sont données en figure 8a et 8b.

Ce programme, implanté à partir de l'adresse 0066_H commence par décrémenter le compteur A' de lignes vides. Quand celui-ci est non nul, la partie vide de l'affichage n'est pas terminée et le programme s'achève par l'instruction RETN afin de permettre la continuation de l'exécution du programme qui a été interrompu par $\overline{NMI}$.

Quand A'=0, l'exécution se poursuit en NMI001 où le contexte du microprocesseur est sauvegardé. Si le test situé aux adresses 0070H et 0074H indique que l'on vient de terminer la partie vide basse de l'écran, un saut est fait en NMI002 où les interruptions sont inhibées et où le registre A' et l'indicateur sont réinitialisés afin de pouvoir gérer la partie vide haute de l'écran lors de la prochaine interruption $\overline{NMI}$. La gestion du clignotement du curseur est ensuite effectuée avant de produire le signal de synchro-trame des adresses 00B2H à 00BBH. Puis la variable FRAMES, utilisée par la fonction PAUSE du BASIC, est

décrémentée et est donc modifiée toutes les 20 ms. Le contexte du microprocesseur est ensuite restitué avant d'autoriser à nouveau les interruptions $\overline{NMI}$ et de terminer le programme.

Il est à noter que le module de gestion du clignotement du curseur dure toujours le même temps, que le bit de poids fort, significatif de la vidéo, soit complémenté ou non. Nécessaire pour une stabilité de l'affichage de l'image sur l'écran, ce temps constant est assuré par des retards produits en CURCL1 des adresses 00D2H à 00D4H.

Lors des 62 (=3EH) interruptions $\overline{NMI}$, pendant lesquelles la SYNCHRO LIGNE est assurée par SLM, le programme s'exécutera jusqu'à l'adresse 006BH . A la 62ème , la détection de A'=0 provoque le saut en NMI001. La valeur du bit 3 de ECRIMP positionné auparavant permet de ne pas sauter en NMI002 et de continuer en séquence où A' est réinitialisé par 39 (=27H) pour la partie basse de l'écran. Afin d'assurer une parfaite synchronisation des signaux SLL et SLM, l'instruction HALT stoppe le microprocesseur qui ne recommencera l'exécution qu'après une interruption $\overline{NMI}$ (qui provoquera A'=38) . Le signal de SYNCHRO LIGNE est généré simultanément grâce à SLM.

Cette reprise a lieu à l'adresse 007AH à partir de laquelle les interruptions $\overline{NMI}$ et le signal SLM sont inhibés, l'indicateur est positionné et le contexte sauvegardé. Un retard de 41T est généré afin d'assurer une parfaite synchronisation entre les signaux SLL et SLM. L'affichage de la partie utile est faite par l'appel du module DISPLA puis les interruptions sont autorisées, le clavier est lu, le contexte est restitué et le programme se termine.

Les 38 lignes vides du bas de l'écran sont générées avec une SYNCHRO LIGNE automatique issue de SLM et les interruptions sont autorisées avant de recommencer à générer une synchro trame.

Le schéma de la figure 9 montre l'adressage du clavier par 8 lignes A8 à A15 du bus d'adresses B02 via un réseau de diodes afin de fournir le bus B23 (figure 1). Cinq informations sont transmises sur les bits D0 à D4 du bus de don-

nées B03 quand l'équation $\overline{IORQ} + \overline{RD} + A0 = 0$ est vérifiée par la porte OU 901 afin de sortir les portes de CI22, bloc B21, de l'état haute impédance.

La scrutation du clavier est faite en imposant un bit à la fois au niveau bas de A15 à A8 et en lisant sur le bus de données si une information 0 existe, ce qui se produit quand une ligne Ai à 0 est mise en contact avec une ligne Dj par l'appui d'une touche. Si aucune touche n'est appuyée, les résistances RR3 imposent un niveau logique 1. Cette scrutation est assurée par le programme SCRCLA de la figure 10. A l'adresse 0144H, le registre C est initialisé par la valeur FEH afin d'avoir A0=0 pour qu'à l'instruction IN A,(C) l'équation $\overline{IORQ} + \overline{RD} + A0 = 0$ soit vérifiée. Lors de cette instruction, le contenu du registre B est présenté sur les bits A15 à A8 du bus d'adresses. A l'initialisation, B=FEH, ainsi les touches N, U, Q, A et shift sont scrutées. Si l'une d'entre elles est appuyée, le bit correspondant du registre A est positionné à 0. Après avoir imposé les trois bits de poids forts inutilisés à 1 par l'instruction OR E0, un test est fait afin de savoir si l'un des bits lus est à 0. Dans la négative, la ligne suivante est scrutée et ce, jusqu'à la huitième ligne sauf si une touche est appuyée. Dans ce cas, le SHIFT peut être détecté en SCRCL1 et mémorisé dans le registre D. Le programme se termine dès qu'une touche est détectée ou que toutes les lignes sont scrutées.

La valeur des registres B et C est alors significative de la touche appuyée et la colonne S des figures 11a et 11b permet de connaître la valeur ainsi calculée selon un clavier de type QWERTY ou AZERTY. Le programme DECCLA de la figure 12 permet, à partir de ce code de 16 bits, de générer en B un code compris entre les valeurs 00 et 4FH incluses, soit 80 valeurs contigues représentant les 40 touches seules, les 39 touches avec SHIFT et aucune touche. Le code calculé est indiqué en colonne D des figures 11a et 11b.

Le programme SCRCLA est tel que le bit 5 du registre C de CI8 est positionné à 0 si et seulement si la touche SHIFT a été appuyée. Le programme DECCLA teste ce bit et initialise un registre D par la valeur 0 si SHIFT n'est pas appuyée ou

par la valeur 28H (=40) dans le cas contraire. Un nombre est fabriqué dans le registre A selon la colonne de la touche appuyée car ce registre, initialisé à 0, est incrémenté tant qu'une valeur 0 n'est pas trouvée dans les bits successifs de C. Si aucune touche n'a été appuyée, ce qui se détecte par A = 5, la valeur 0 est mise en B pour l'indiquer avant de terminer le traitement.

Dès qu'une colonne est détectée, un saut est fait en DECCL2 pour chercher la ligne de la touche après avoir sauvegardé le nombre significatif de la colonne dans un registre E. Un nombre est ensuite produit par incrémentation du registre A jusqu'à ce qu'un bit 0 soit trouvé. Quand la ligne a été trouvée, un nombre est calculé en B, compris entre 00 et 4FH, qui tient compte de la colonne, de la ligne et du SHIFT.

Le programme CARMOD, dont l'organigramme est présenté en figures 13a,b,c s'exécute à la suite de SCRCLA et DECCLA afin de fournir un code significatif de la touche appuyée. Chaque touche du clavier du micro-ordinateur objet du présent brevet peut avoir jusqu'à huit interprétations différentes dépendantes du mode de traduction dans lequel le système se trouve lors de l'appui d'une touche. Une même touche peut fournir un caractère alpha-numérique en majuscule ou en minuscule, qui peuvent éventuellement avoir leur vidéo inversée, deux types de fonctions différentes et, lorsque la touche SHIFT est aussi appuyée, une autre fonction ou un caractère graphique ; les fonctions disponibles sont soit des fonctions du langage BASIC, soit des ordres spécifiques au micro-ordinateur tels l'inversion vidéo, l'accès au langage machine, la sélection du mode d'affichage sur l'écran, les fonctions d'édition et de déplacement du curseur dans les quatre directions, la sauvegarde sur cassette, la sélection des majuscules ou des minuscules, la sélection du mode graphique...

Une variable MODACT, figure 7, permet d'indiquer dans quel mode la touche détectée doit être interprêtée par CARMOD. Les colonnes 1, 2, 3 et 4 des figures 11a et 11b indiquent les résultats de la traduction en fonction des modes de même numéro. En mode 1, la touche est comprise

comme une instruction ou un numérique. Le mode 2 est réservé à la traduction de la touche en caractère alphanumérique majuscule ou minuscule selon la valeur du bit 7 de MODACT. Le mode 3 donne le code d'une fonction. Et enfin, le mode 4 sert à fournir le code de caractères graphiques, majuscules ou minuscules. Le mode 5 est réservé pour les fonctions d'aides au travail en langage machine.

Le programme CARMOD interprète le code fourni en B par DECCLA selon le bit mis à 1 dans MODACT, qui est significatif du mode sous lequel la touche doit être interprêtée, afin de générer un nouveau code en B et un indicateur en C qui prendra la valeur 00 pour un caractère affichable, la valeur 8D pour une fonction et la valeur 01 dans les autres cas. La valeur du code mis en B est reproduit dans les colonnes b des figures 11a et 11b et a pour signification ce qui est dans les colonnes a. CARMOD indique également, colonne c, le mode sous lequel la prochaine touche devra être interprêtée.

L'organigramme de la figure 13a montre qu'une table, indiquée en figure 14, est utilisée et est adressée par le registre HL du microprocesseur afin de fournir le code recherché. Cette table est reproduite en figure 14. Le registre H est initialisé par une valeur constante, 1BH, égale aux huit bits de poids forts de l'adresse de cette table alors que le registre L variera selon la touche et le mode. Le registre C est initialisé à 0 afin d'indiquer, par hypothèse, que le code donnera un caractère affichable. Cette valeur pourra éventuellement être modifiée par la suite. MODACT est ensuite testé pour savoir sous quel mode il faut interprêter la touche appuyée.

Quand il faut traduire sous le mode 4, mode graphique, le code de la touche fourni par DECCLA est mis en L afin d'adresser par HL un nouveau code dans la table de la figure 14 et le ranger en E. Par exemple, si la touche D a été appuyée, la figure 11a montre que SCRCLA fournit BC=FDFDH et DECCLA produit B=09. le programme CARMOD va alors chercher le contenu de l'adresse 1B09H (figure 14) qui contient la valeur 44H, c'est à dire le code ASCII de la lettre "D". Dans le cas où le code trouvé dans la table le la figure 14 est celui de la touche ENTER, avec ou sans SHIFT, le registre

C est positionné à 1 afin d'indiquer qu'aucun code n'est à prendre en compte et les variables MODACT et MODANC sont réinitialisées au mode 2 car l'appui de cette touche ne sert qu'à sortir du mode graphique. La variable MODANC sera utilisée ultérieurement.

Si la touche EFFACE est appuyée, la fonction est indiquée par 8D.

Dans le cas de touches appuyées simultanément avec SHIFT, le code est directement fourni par E et correspond à l'un des graphismes de la colonne 4a de la figure 11b. Les touches sans appui simultané de SHIFT correspondent à des caractères alpha-numériques et le mode graphique consiste à inverser la vidéo en positionnant le bit 7 de E à 1 (colonne 4a de la figure 11a). Dans le cas où le caractère est alphabétique et que le mode minuscule est indiqué par le bit 7 de MODACT, la valeur 20H est ajoutée à E. Le code est ensuite rangé en B avant de donner à la variable MODANC la même valeur que MODACT.

Quand la touche est interprêtée selon le mode 3, figure 13b, l'appui éventuel de SHIFT est ignoré car seule une fonction peut être détectée. Un nouveau code est acquis grâce à son adresse fournie par HL où L est l'octet fourni par DECCLA. si une touche sans fonction est appuyée, C=1 et le mode futur est réinitialisé par le mode précédent qui a été sauvegardé auparavant en MODANC. Quand une touche valide de fonction est détectée, C=8D pour l'indiquer et le code réel de la fonction s'obtient en rajoutant la valeur 80H à E. Si le code est celui de la fonction GRAPHIQUE, C=1 et MODACT est initialisé par le mode 4. Si la touche qui permet de faire basculer les choix MAJUSCULE et MINUSCULE est détectée, les bits 7 de MODANC et MODACT sont complémentés et C=1. dans les autres cas, le mode 2 est sélectionné et le code est rangé en B.

Quand la touche est interprêtée selon le mode 1, figure 13c, un test est fait pour savoir si la touche SHIFT a été actionnée simultanément. Dans la négative, le code est pris en table grâce à HL où L contient le code fourni par DECCLA. Si ce code correspond à un chiffre, C n'est pas positionné à 8D car les touches contenant les chiffres ne

comportent pas de fonctions interprêtables sous le mode 1 comme c'est le cas des autres touches qui provoquent alors MODACT= Mode 2. Le code est ensuite rangé en B et, dans le cas où le mode 5 avait été auparavant choisi, les modes 2 et 5 sont imposés en MODACT. C est positionné à 1 sauf dans le cas des fonctions EFFACE et de déplacements horizontaux du curseur.

Si la touche SHIFT a été appuyée simultanément, elle est supprimée et un code pointé par HL est acquis en E. Si ce code correspond à une touche où une fonction est implantée alors C=8D et un nouveau code pointé par HL est pris. Si ce code est celui des fonctions EDIT ou THEN le mode 1 est positionné en MODACT. Si le code ne correspond pas aux fonctions de déplacements verticaux du curseur, le mode 2 est choisi pour le futur. Si le code trouvé ne correspond pas à une touche avec fonction, le caractère est alors codé en ASCII en ajoutant 22H à E afin d'adresser par HL un nouveau code en table sauf pour les touches ESPACE ou ENTER qui, pour cette dernière, provoque MODACT = Mode 3 et C=1. Par exemple, l'appui de SHIFT N provoque (se reporter figure 11b) BC=FECFH par SCRCLA et B=48H par DECCLA. En enlevant la valeur 28H, B est ramené à 20H afin de pouvoir lire le contenu de la case 1B20H, c'est à dire (figure 14) le code 4EH. Cette touche ne correspondant ni à une fonction, ni à ESPACE ou ENTER, 22H est ajouté à ce code pour obtenir ainsi l'adresse 1B70H qui contient le code ASCII de "<", soit 3CH.

Lorsque le mode 2 ou 5 a été sélectionné et que la touche SHIFT est appuyée simultanément, le traitement est identique à cette même configuration en mode 1. Sinon une adresse est produite grâce à B afin de prendre un code en table. Si celui ci correspond à un caractère alphabétique et que le mode minuscule est positionné, la valeur 20H est ajoutée afin d'obtenir la valeur ASCII exacte. Si la touche ENTER a été appuyée, le mode 1 est choisi.

Lorsque, en fin d'exécution du programme CARMOD, le registre C contient la valeur 0, le contenu du registre B est le code d'un caractère affichable correspondant à une touche du clavier. Ce code est celui utilisé lors de l'affi-

chage vidéo pour adresser la matrice du caractère correspondant comme celà a été expliqué auparavant selon la figure 3.

Une procédure non expliquée dans le présent brevet permet, grâce au registre C=0, de reconnaître les caractères affichables afin de les ranger dans la zone mémoire image de l'écran décrite précédemment. Quand C=8D, une fonction est détectée et peut être exécutée immédiatement. Ces fonctions peuvent être soit des utilitaires, soit des fonctions ou commandes du langage BASIC. Quelqu'en soit la nature, elles sont donc toujours déclenchées matériellement par l'appui d'une touche du clavier. Les fonctions implantables sont multiples et les figures 11a et 11b indiquent une disposition possible sur les touches du clavier. On peut remarquer la fonction GRAPHIQUE (code 8DB5) permettant de passer en Mode 4, la fonction MAJmin (code 8DB7) pour sélectionner les caractères en majuscule ou en minuscule, la fonction VIDEO qui commande la bascule CI14 de la figure 3, et la fonction MACHINE (code 8DBO) qui sert à reconfigurer les fonctions du clavier de sorte que l'appui de certaines touches permette de travailler en hexadécimal et d'examiner des zones de mémoire, des registres, de positionner des points d'arrêt, d'exécuter des programmes écrits en langage machine, etc... La fin de l'exécution de ces fonctions utilisées en tant que commande, c'est à dire non incluses dans une suite d'instructions de programme écrit en BASIC, provoque l'affichage à l'écran d'un message en français indiquant à l'utilisateur le résultat des opérations demandées. Il apparaîtra aisément au lecteur que la place des fonctions sur les touches et les libellés des messages peuvent être modifiés sans compromettre le fonctionnement des principes inventifs objets du présent brevet.

La mémoire vive RAM est organisée en plusieurs zones d'informations dont certaines ont leurs positions fixes et d'autres peuvent varier selon leur taille. Trois pointeurs pour chaque zone variable déterminent l'adresse de début, l'adresse de fin et l'adresse contenant l'octet sur lequel

une opération peut être effectuée. Comme l'effacement de l'une de ces zones se fait en rendant égales les adresses contenues par ces trois pointeurs, il se peut que deux zones qui étaient consécutives deviennent séparées par une zone vide, appelée tampon, lorsque l'une de ces deux zones est effacée. La première zone qui nous intéresse dans la présente explication est réservée pour stocker des programmes ou des données écrits en langage Machine grâce à la fonction MACHINE vue précédemment qui positionne en Mode 5. Son début est fixe et est situé à l'adresse DEBUSR=445OH Cette zone est suivie par des zones d'adresses variables dont, par ordre d'adresses croissantes, celles pour ranger les programmes écrits en BASIC, les variables déclarées par les instructions du BASIC et les données utilisées par les instructions READ et DATA du BASIC dont l'adresse de fin de zone est contenue dans une variable appelée FINDAT. Ces quatre zones sont sauvegardables sur cassette ainsi que l'ensemble des pointeurs contenant leurs adresses.

L'utilisation d'un magnétophone à cassette en tant que mémoire de masse implique d'avoir une liaison entre le micro-ordinateur et l'entrée MICRO du magnétophone afin de pouvoir enregistrer les informations sur une cassette. La récupération de cette information est assurée par une liaison du micro-ordinateur avec la sortie ECOUTEUR ou HAUT-PARLEUR de ce magnétophone. Ces liaisons et le décodage et codage des informations sont assurés par le matériel dont une partie est contrôlée par des instructions binaires situées en mémoire morte.

La sauvegarde des informations se fait par la détection de la touche SAVE par CARMOD qui fournit C=8D et B=42 (figure 11a). L'organigramme de la figure 15 montre le déroulement des opérations nécessaires à cette sauvegarde. Afin de ne pas être perturbé par les opérations d'affichage vidéo, la première opération consiste à inhiber les interruptions $\overline{\text{NMI}}$ grâce à une instruction OUT (OFD),A dont l'action sur le matériel a déjà été décrite lors de l'explication de la figure 4. Il a été également vu lors de ces explications que cette instruction sert également à sélectionner les signaux de synchro pilotés par logiciel, or le schéma de la figure

16, détail du bloc B20 de la figure 1, montre que le signal destiné à la cassette est issu de la sortie SYNCHRO-LIGNE après passage dans un filtre passe-bande dont les fréquences de coupures sont de 1200 et 2400 Hz. Ce filtre est réalisé par les résistances R30 et R31 ainsi que les condensateurs C6 et C7 et est tel que l'impédance de sortie soit de 1000 Ohms et que le signal ait un niveau de 5 mV afin de pouvoir être connecté à l'entrée MICRO d'un magnétophone. Le signal de nom SYNCHRO-LIGNE n'a pas les mêmes caractèristiques que pour un affichage sur l'écran car il est constitué de trains d'impulsions de 1200 et 2400 Hz obtenus par un procédé expliqué par la suite.

Le second bloc logique de l'organigramme de la figure 15 sert à supprimer les tampons éventuels afin qu'il n'y ait que l'ensemble des octets utiles à sauvegarder, c'est à dire les adresses définissant chaque zone et les quatre zones citées auparavant. L'absence du nom de programme qui doit normalement suivre l'instruction SAVE provoque un branchement à un programme indiquant une erreur. Si la taille du nom du programme est supérieure à 255 caractères, une erreur est également produite.

L'envoi du signal représentant un bit "1" pendant cinq secondes afin de faciliter la relecture de la cassette précède l'envoi du nom du programme et de l'adresse de fin de zone à sauvegarder : FINDAT. Un pointeur est ensuite positionné afin de pointer l'adresse du premier octet à sauvegarder puis les octets sont envoyés un à un en incrémentant ce pointeur. Dès que celui-ci dépasse l'adresse de fin de zone sauvegardable, le mode principal d'affichage est restitué en autorisant éventuellement les interruptions $\overline{NMI}$ et terminant ainsi la sauvegarde.

Une temporisation est faite afin qu'il n'y ait aucun temps mort entre l'envoi de deux octets et que la synchronisation des signaux soit parfaite entre le dernier bit STOP d'un octet et le bit START de l'octet suivant. Cette synchronisation est assurée par le signal SYNCHRO imposé au niveau bas pendant 683 périodes d'horloge T.

Un octet est envoyé vers la cassette en le faisant précéder d'un signal START, symbolisé par un bit 0, pour

signaler sa venue et en le faisant suivre d'un bit de parité, dont la cohérence sera vérifiée à la relecture, et de deux signaux STOP, symbolisés par des bits 1, afin de retrouver plus facilement le bit START de l'octet suivant.

Le bit 0 est caractèrisé par un groupe de 4 impulsions, figure 19, dont la durée au niveau haut, identique à la durée au niveau bas, est de 1366 T afin d'avoir une fréquence de l'ordre de 1200 Hz. Le bit 1 est caractèrisé par un train de huit impulsions, figure 20, dont les niveaux haut et bas durent chacun 683 T afin de produire une fréquence de l'ordre de 2400 Hz. Les organigrammes de ces sous-programmes contrôlant ces durées et ces niveaux sont reproduits en figures 17 et 18.

Avant l'entrée dans le programme de génération d'un bit 1, figure 18, le registre B doit être initialisé par une valeur permettant de synchroniser le nouveau bit par rapport au bit précédemment envoyé. Cette valeur est telle que l'on ait un retard de 1366 T avant le passage au niveau haut si le bit précédent était un 0 et 683T si le bit précédent était un 1. Le registre C est initialisé par le nombre d'impulsions à envoyer, c'est à dire 8. Après la boucle d'attente à l'adresse 2079H permettant d'ajuster les temps, l'accumulateur A est chargé par la valeur 7FH afin d'adresser la ligne du clavier qui contient la touche "ARRET" lorsque l'instruction IN A,(OFE) sera exécutée (se reporter à l'explication de la figure 9).

Cette instruction permet de lire le clavier afin de savoir si la touche "ARRET" a été appuyée. dans l'affirmative, les instructions RRA et JP NC,ARRE16 le vérifient et provoquent un branchement à une routine de traitement de l'arrêt. Cette instruction IN A,(OFE) s'accompagne toujours de $\overline{RD}$=0, $\overline{WR}$=1, A0=0 et $\overline{IORQ}$=0. L'état de ces signaux est vérifié par les portes 416, 428 et 429 et la figure 4 montre que cette configuration bloque la bascule RS, constituée de 430 et 431, de sorte que l'entrée CLR de CI20 soit à 0, forçant ainsi le signal SLL au niveau bas en permanence et par conséquent, puisque les signaux contrôlés par logiciel sont sélectionnés, le signal SYNCHRO-LIGNE à 1.

Les trois instructions ADD A,(IY+0), LD B,31 et DJNZ EUN3 ne servent qu'à ajuster précisemment le temps à 683 T avant de faire revenir le signal SYNCHRO à 0 par l'instruction OUT (0FFH),A car, comme le montre la figure 4, les signaux simultanés $\overline{IORQ}$=0, $\overline{RD}$=1 et $\overline{WR}$=0 accompagnant cette instruction provoquent la mise au niveau haut de l'entrée CLR de CI20. Ainsi, la sortie SLL repassera à 1 après que deux instructions soient recherchées pendant lesquelles les fronts descendants de $\overline{M1}$ excitent les entrées CK de CI19 et CI20 afin de propager le niveau imposé par la sortie $\overline{Q}$ de CI18 jusqu'à la sortie. Ce basculement de SLL provoque le passage au niveau bas de la sortie SYNCHRO-LIGNE.

Le compteur C du nombre de créneaux envoyés est décrémenté jusqu'à 0 indiquant ainsi la fin de la routine. Tant que ce compteur n'atteint pas la valeur 0, un retard afin d'ajuster à 683 T est fait par l'ensemble des instructions qui suivent avant de reprendre la génération d'un signal SYNCHRO-LIGNE = 1.

L'envoi d'un bit 0 est fait exactement selon le même principe. La figure 17 indique que le registre C est chargé par la valeur 4, correspondant au nombre d'impulsions à envoyer puis la touche "ARRET" est testée en faisant passer SYNCHRO-LIGNE à 1. Après une temporisation à l'adresse 2062H pour ajuster à 1366 T, le signal SYNCHRO-LIGNE est imposé à 0 par OUT (0FF),A et, si le registre C ne prend pas la valeur 0 après décrémentation, un retard afin d'ajuster à 1366 T est provoqué avant de reprendre la génération d'un niveau 1.

La durée du niveau bas des signaux de la dernière impulsion envoyée n'est pas contrôlée implicitement par le programme car celui-ci ne produit en fait que 734 T pour la création d'un bit 0 et 39 T pour un bit 1. Ces durées seront prises en considération par le module qui envoi l'octet en entier afin de respécter entre deux bits les temps respéctifs de 1366 T ou 683 T avant de faire SYNCHRO-LIGNE = 1.

On voit qu'on a décrit ci-dessus un micro-ordinateur offrant des possibilités d'affichage vidéo, de lecture de clavier comportant des touches multi-fonctions et de liaison avec un magnétophone pour utiliser celui-ci comme mémoire de masse. Le micro-ordinateur est fabriqué à partir d'articles disponibles dans le commerce mais peut aussi justifier la fabrication d'un composant spécifique regroupant l'ensemble ou une partie des fonctions décrites en un seul circuit. L'utilisation de ressources déjà existantes dans un foyer et l'accès aisé des fonctions préprogrammées du langage BASIC ainsi que des fonctions d'aide au travail en langage machine en font un excellent outil d'enseignement et d'initiation à l'informatique. De plus les principes inventifs décrits dans le présent brevet permettent de fabriquer cet appareil à un faible coût, le rendant ainsi accessible pour la plupart des foyers.

REVENDICATIONS

1) Micro-ordinateur caractèrisé en ce qu'il comporte une unité centrale de traitement, une mémoire vive (mémoire RAM), une mémoire morte, une interface de liaison avec un clavier, une interface de sortie vers un magnétophone, une interface de génération de signaux de synchro et de vidéo destinés à un écran à tube cathodique et une logique de génération de signaux d'horloge destinés à ladite unité de traitement et auxdites interfaces; ladite unité centrale de traitement comprenant un bus de données connecté auxdites interfaces et, à travers des résistances, auxdites mémoires RAM et mortes ainsi qu'à un moyen de type registre pour mémoriser des signaux issus de ladite mémoire RAM; ladite mémoire morte contenant des instructions et des données codées sous forme binaire afin de contrôler lesdites interfaces et les fonctions qui en découlent; ladite mémoire morte étant connectée à des multiplexeurs de telle manière qu'elle puisse être adressée soit par un bus d'adresses provenant de l'unité centrale de traitement afin que celle-ci puisse lire les informations contenues dans cette mémoire, soit par un ensemble de signaux destinés à lire une information utilisée par l'interface vidéo; lesdits multiplexeurs étant commandés par un signal issu de ladite unité centrale de traitement; ledit signal vidéo étant produit par une information mémorisée dans un registre à décalage et dont le décalage est commandé par des impulsions.

2) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que la mémoire morte d'une capacité totale de 16 Kilo-octets implantée à partir de l'adresse 0000 peut être constituée de un ou plusieurs circuits standards du commerce de telle sorte qu'un même emplacement réservé peut recevoir indifféremment une mémoire morte de 16 Kilo-octets ou une mémoire morte de 8 Kilo-octets; la sélection du type de circuit choisi se faisant grâce à des liaisons à établir entre certains points.

3) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que la ligne A13 du bus d'adresses issu de l'unité centrale de traitement est reliée à l'entrée d'une

porte OU, elle-même connectée à une entrée d'un multiplexeur dont la sortie excite une entrée de sélection de ladite mémoire morte si celle-ci est de capacité 8 Kilo-octets et caractèrisé par le fait que ladite entrée de ladite porte OU est imposée à un niveau constamment bas si ladite mémoire est de capacité 16 Kilo-octets.

4) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que l'ensemble des signaux d'adressage de ladite mémoire morte, lorsque lesdits multiplexeurs ne sélectionnent pas ceux issus dudit bus d'adresses de l'unité centrale de traitement, sont constitués par trois bits d'un compteur par huit, par sept des huit bits dudit moyen de type registre contenant le code représentatif d'un caractère affichable, et par six bits prédéfinis afin d'adresser dans ladite mémoire une case comprise entre les adresses hexadécimales 1C00H et 1FFFH incluses.

5) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que tout octet peut représenter le code d'un caractère affichable sauf le code héxadécimal FFH.

6) Micro-ordinateur suivant la revendication 5 caractèrisé par le fait que la détection de tout code représentatif d'un caractère affichable, pendant la recherche d'une instruction par l'unité centrale de traitement, impose à ladite unité centrale de traitement la lecture et l'exécution d'une instruction ne faisant aucune opération; cette détection permettant également de mémoriser dans un registre ledit code afin d'adresser ultérieurement la mémoire morte.

7) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que lesdites impulsions sont issues de la sortie Q d'une bascule de type D dont l'entrée CLR active au niveau bas est reliée à la sortie $\overline{Q}$ et dont l'entrée D est imposée au niveau haut; une impulsion étant produite à chaque front montant imposé à l'entrée d'horloge CK de cette bascule.

8) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que le signal vidéo est prédéfini par la sortie d'une bascule initialisée lors de la mise en fonctionnement dudit micro-ordinateur et est inversé par la détection de signaux prédéterminés issus de l'unité centrale de traitement.

9) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que le signal de synchro-ligne destiné à un écran provient soit de signaux produits uniquement par des circuits logiques, soit par des signaux produits par des circuits logiques contrôlés par des informations binaires contenues dans la mémoire morte.

10) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que ledit signal de synchronisation est à un niveau logique défini pendant 15 périodes d'une horloge et à un niveau opposé pendant 193 périodes de la même horloge.

11) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que l'écran est divisé de manière telle qu'il comprend une partie constituée de 63 lignes pour lesquelles le signal de synchro-ligne est produit par les circuits logiques seuls, une partie constituée de 209 lignes pendant lesquelles le signal de synchro-ligne est contrôlé par des informations contenues dans la mémoire morte, et pendant lesquelles un signal de vidéo est envoyé pendant les 208 premières lignes, et enfin une troisième partie composée de 38 lignes pendant lesquelles le signal de synchro-ligne est produit par les circuits logiques seuls.

12) Micro-ordinateur suivant la revendication 11 caractèrisé par le fait que les parties pendant lesquelles ledit signal de synchro-ligne est produit par les circuits logiques seuls sont contrôlées par un indicateur afin de savoir laquelle de ces deux parties est en cours et caractèrisé par le fait que le nombre de lignes composant chacune de ces parties est décompté par un compteur de type registre.

13) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que ladite mémoire RAM contient les codes de caractères affichables de telle manière que ceux-ci sont contigus et que le dernier code d'une ligne d'écran est toujours suivi du caractère spécial non reconnu comme le code d'un caractère affichable.

14) Micro-ordinateur suivant la revendication 5 caractèrisé par le fait que le code ne représentant pas un caractère affichable provoque l'exécution, à une adresse implicitement prédéfinie par ledit code, d'une suite d'instructions destinées à gérer l'affichage sur un écran.

15) Micro-ordinateur suivant la revendication 14 caractèrisé par le fait que ladite suite d'instruction consiste à produire un signal de synchro-ligne contrôlé par la mémoire morte et à produire le signal de vidéo; l'ensemble de ces instructions étant construit de manière telle que ledit signal de synchro-ligne est identique à celui produit par des circuits logiques seuls et organisé de manière telle que les cumuls des temps d'exécution desdites instructions sont identiques quelques soient les décisions prises lors de l'exécution de l'ensemble des instructions avant de produire le signal vidéo.

16) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que l'appui d'une touche dudit clavier produit la génération d'un code hexadécimal dans un moyen de mémorisation de type registre; ledit code ayant pour valeur 00 lorsqu'aucune touche n'est appuyée ou lorsque la touche SHIFT seule est appuyée, 1F pour la touche ENTER, 07 pour ESPACE, 17 pour le ".", 0F pour 0, 16 por 1, 0E pour 2, 06 pour 3, 14 pour 4, 0C pour 5, 04 pour 6, 15 pour 7, 0D pour 8, 05 pour 9, 08 pour A pour un clavier QWERTY ou pour Q pour un clavier AZERTY, 21 pour B, 02 pour C, 09 pour D, 11 pour E, 0A pour F, 1A pour G, 23 pour H, 1D pour I, 25 pour J, 24 pour K, 26 pour L, 27 pour M, 20 pour N, 1C pour O, 1E pour P, 10 pour Q pour un clavier QWERTY ou pour A pour un clavier

AZERTY, 12 pour R, 0B pour S, 19 pour T, 18 pour U, 22 pour V, 13 pour W dans le cas d'un clavier QWERTY ou pour Z dans le cas d'un clavier AZERTY, 01 pour X, 1B pour Y, 03 pour Z dans le cas d'un clavier QWERTY ou pour W dans le cas d'un clavier AZERTY; lorsque la touche SHIFT est appuyée simultanément avec une autre touche, ledit code ayant pour valeur 47 lorsque l'autre touche est ENTER, 2F pour ESPACE, 3F pour le ".", 37 pour 0, 3E pour 1, 36 pour 2, 2E pour 3, 3C pour 4, 34 pour 5, 2C pour 6, 3D pour 7, 35 pour 8, 2D pour 9, 30 pour A dans le cas d'un clavier QWERTY ou pour Q dans le cas d'un clavier AZERTY, 49 pour B, 2A pour C, 31 pour D, 39 pour E, 32 pour F, 42 pour G, 4B pour H, 45 pour I, 4D pour J, 4C pour K, 4E pour F, 4F pour M, 48 pour N, 44 pour O, 46 pour P, 38 pour Q dans le cas d'un clavier QWERTY ou pour A dans le cas d'un clavier AZERTY, 3A pout R, 33 pour S, 41 pour T, 40 pour U, 4A pour V, 3B pour W, dans le cas d'un clavier QWERTY ou pour Z dans le cas d'un clavier AZERTY, 29 pour X, 43 pour Y, 2B pour Z dans le cas d'un clavier QWERTY ou pour W dans le cas d'un clavier AZERTY.

17) Micro-ordinateur suivant la revendication 16 caractérisé par le fait que ledit code produit est traduit selon la valeur d'une information mémorisée dans un moyen de type registre ou mémoire afin de fournir un nouveau code, dans un moyen de type registre ou mémoire, significatif du contenu de la touche appuyée quand ladite touche peut avoir plusieurs significations possibles; lorsque la valeur de l'information servant à la traduction est telle que ladite touche doit être traduite comme étant significative d'un type de fonctions, le code produit est un code conforme au standard ASCII pour les touches numériques, la touche ESPACE, la touche ENTER, et la touche ".", pour les autres touches, le code produit est composé de deux octets : le premier d'une valeur constante indiquant qu'il s'agit d'une fonction et le second conforme au code ASCII de la lettre qui se trouve sur la même touche de ladite fonction; quand la touche SHIFT est appuyée simultanément avec une autre touche, le code produit est conforme au standard ASCII s'il existe un caractère sur ladite touche ou bien est constitué de deux octets pour

indiquer la prise en compte d'une fonction : la valeur constante précédemment citée suivie d'une valeur significative de la fonction sélectionnée par ladite touche ; lorsque la valeur de l'information servant à la traduction est telle que la touche doit être traduite comme étant significative d'un caractère, le code produit est conforme au standard ASCII si la touche SHIFT n'est pas appuyée simultanément sinon il est identique au code produit lorsque la touche doit être traduite comme étant une fonction vue précédemment lorsque la touche SHIFT est appuyée ; lorsque la valeur de l'information servant à la traduction est telle que la touche doit être traduite comme étant significative d'une fonction d'un type différent de celui déjà vu précédemment et que la touche SHIFT soit appuyée ou non, le code produit est toujours précédé de la valeur constante déjà citée significative d'une fonction suivie d'un code dont la valeur correspond au code ASCII du caractère également disposé sur ladite touche augmenté de la valeur 80H ; lorsque la valeur de l'information servant à la traduction est telle que la touche doit être traduite comme étant significative d'un caractère graphique, le code produit a pour valeur le code correspondant au caractère ASCII disposé sur ladite touche augmenté de la valeur 80H.

18) Micro-ordinateur suivant la revendication 1 caractèrisé par le fait que le code significatif d'une touche du clavier est codé sur deux octets lorsque ladite touche ne représente pas un caractère alphanumérique.

PLANCHE I/I8

FIG.1

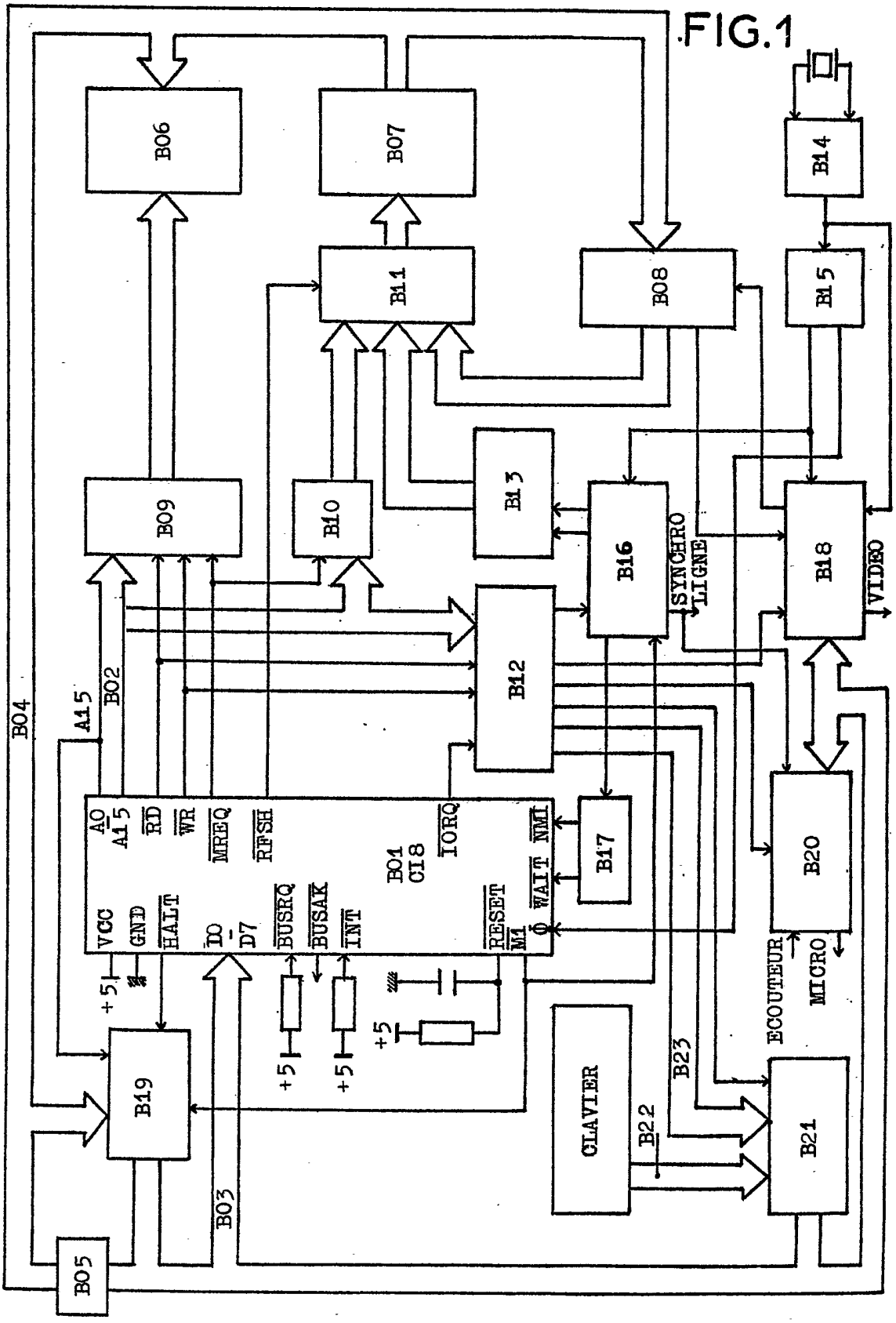


PLANCHE 2/I8

FIG. 2

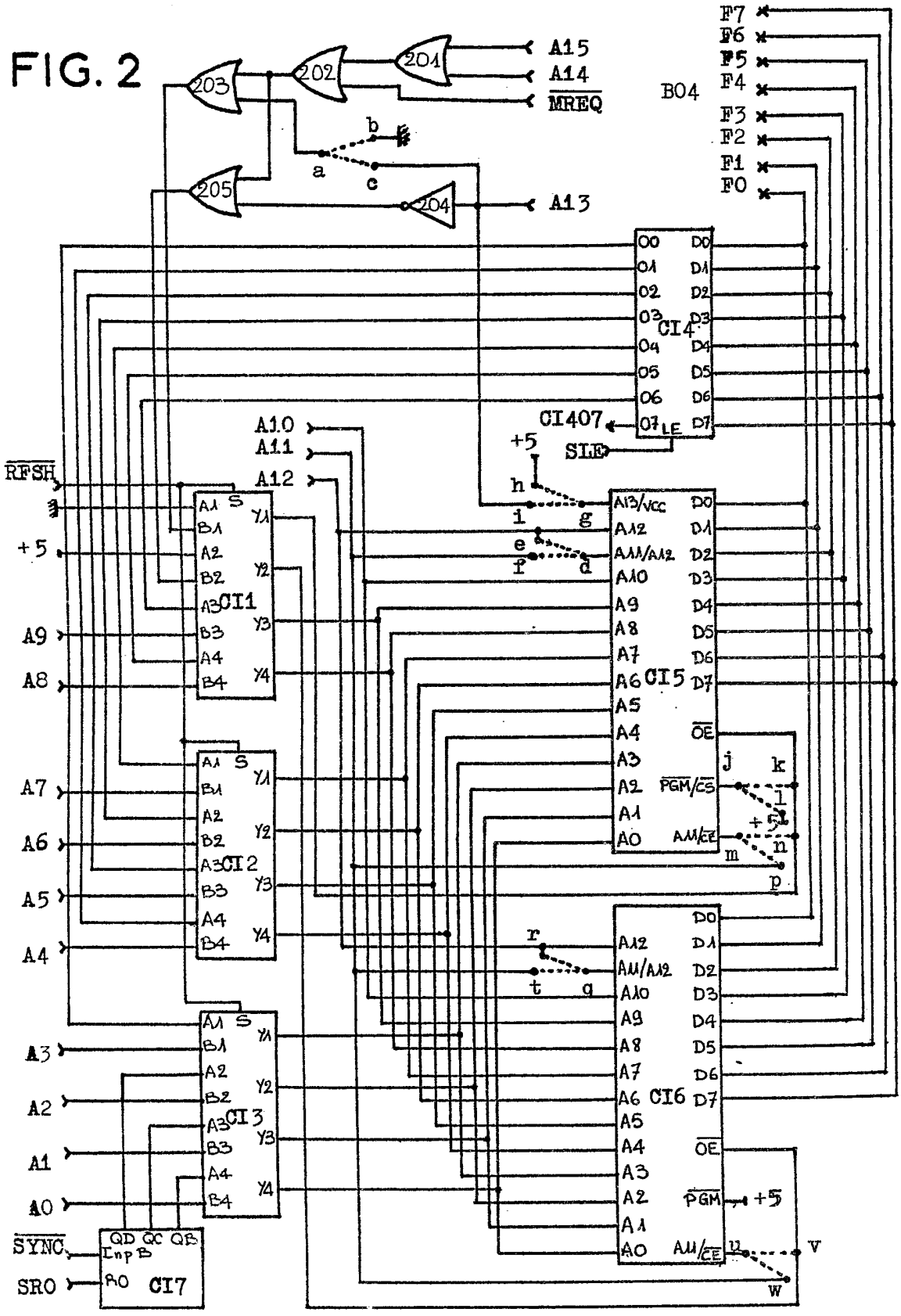


PLANCHE 3/I8

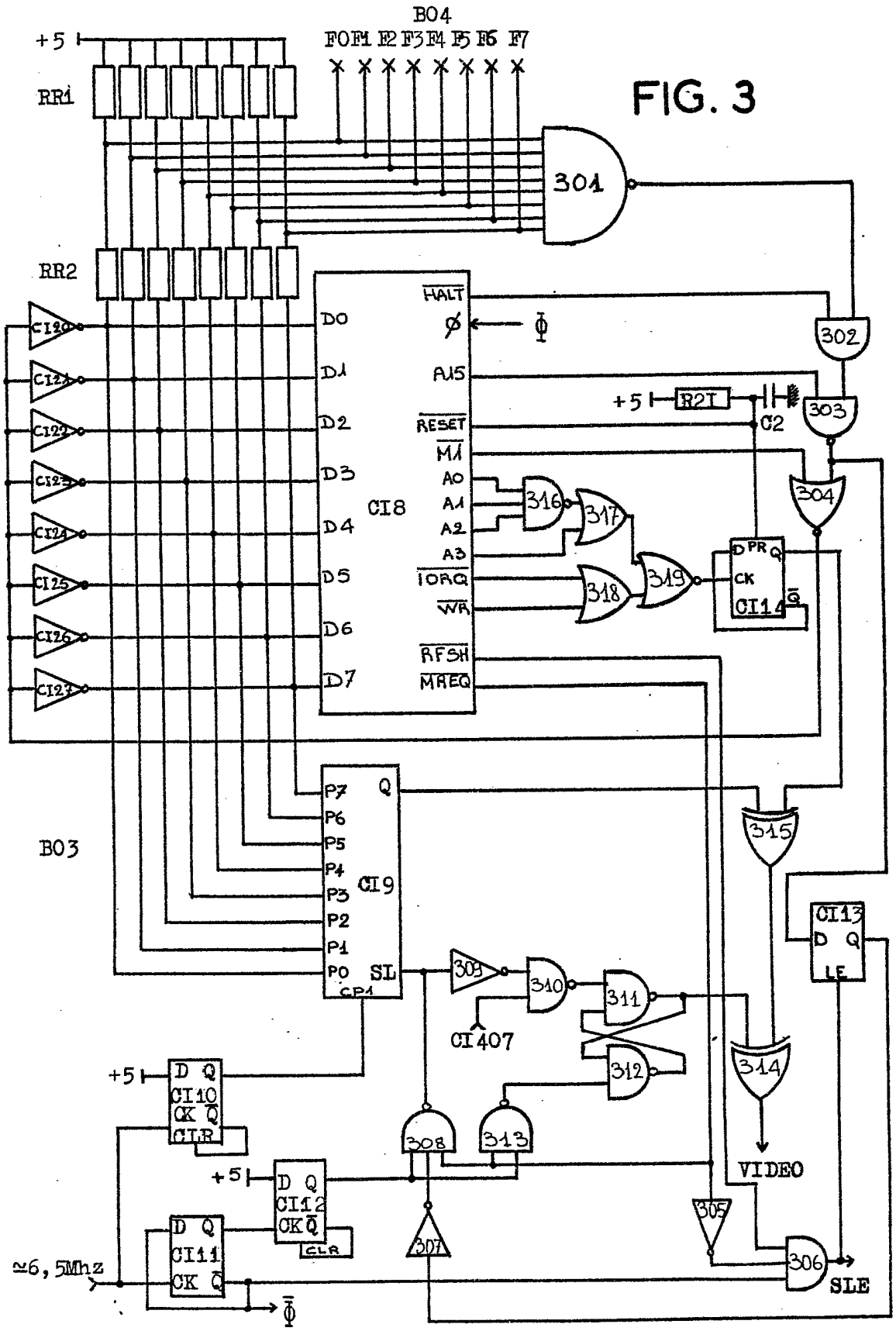


FIG. 3

PLANCHE 4/I8

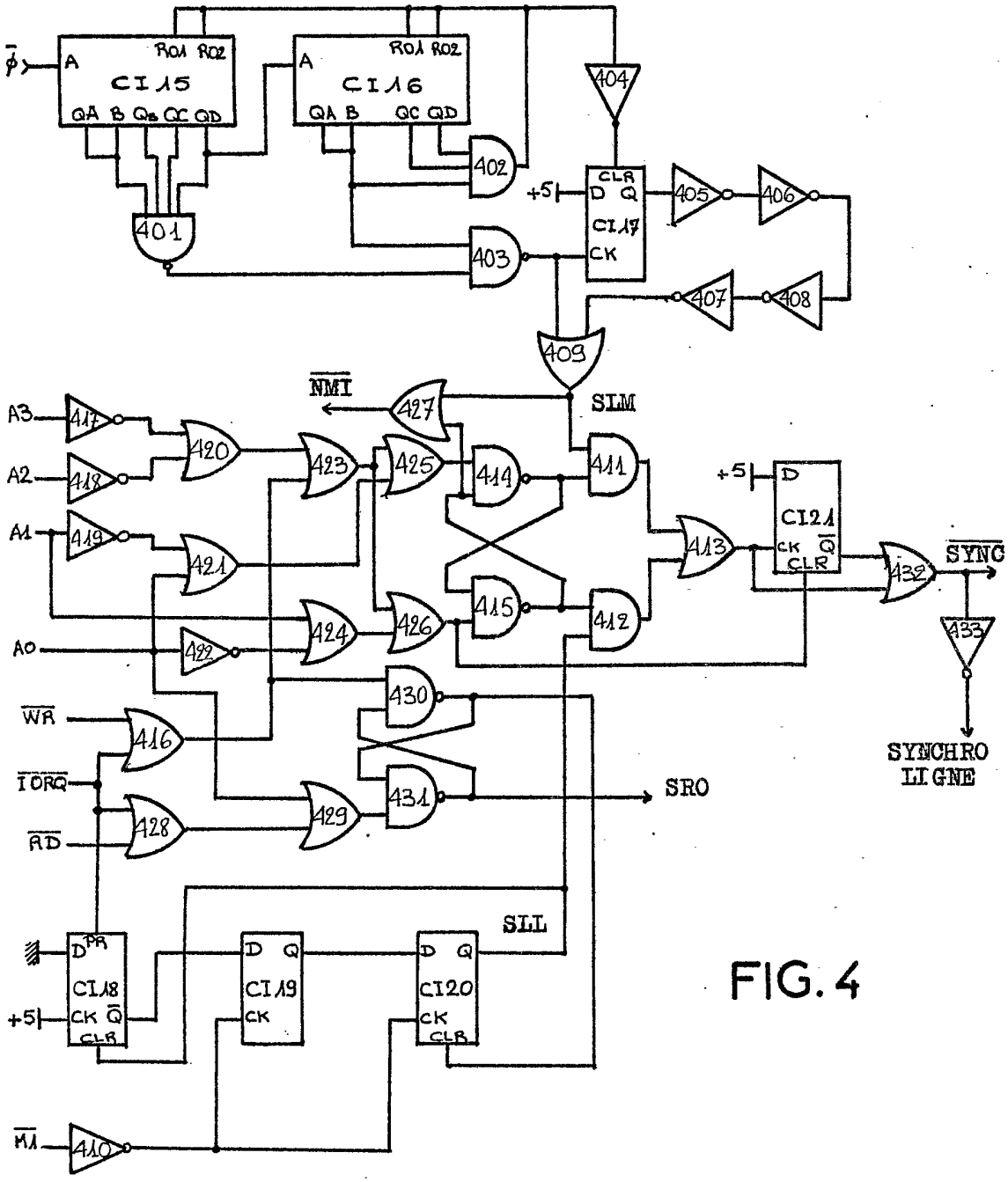


FIG. 4

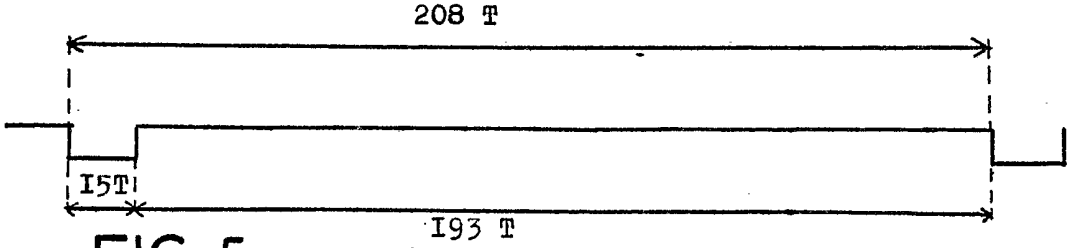


FIG. 5

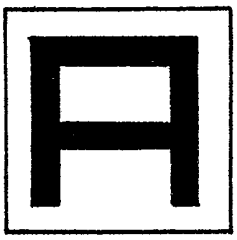
Adresse	Contenu hexadecimal	Contenu binaire	Affichage sur l'ecran
1E08	00	00000000	
1E09	7E	01111110	
1E0A	42	01000010	
1E0B	42	01000010	
1E0C	7E	01111110	
1E0D	42	01000010	
1E0E	42	01000010	
1E0F	00	00000000	

FIG. 6

BITS DE MODACT	INTERPRETATION QUAND LE BIT=1	MODE
7 6 5 4 3 2 1 0		
0	Instruction centrale	1
1	Caractère alpha-numérique	2
2	Instruction du bas	3
3	Caractère graphique	4
4	Fonction machine	5
7	0 = Majuscule 1 = Minuscule	

FIG. 7

PLANCHE 6/I8

```
;GESTION DES INTERRUPTIONS INT-RST38

 0038    D3 FF      INTERR: OUT (0FF),A ;POUR SYNCHRO=0 .11T
 003A    0D                 DEC C       ;LIGNES MATRICE . 4T
 003B    C2 0045            JP  NZ,INTER1;SAUT SI NON 0 .10T
 003E    C0                 RET NZ      ;POUR 15T       . 5T
 003F    E1                 POP HL      ;LIGNE SUIVANTE .10T
 0040    05                 DEC B       ;DECOMPTE SUR 26. 4T
 0041    C8                 RET Z       ;FIN SI DERNIERE. 5T
 0042    CB D9              SET 3,C     ;INIT COMPT.=8  . 8T
 0044               EXELI6:             ;               .
 0044    E9         INTER2: JP  (HL)    ;AFFICHE        . 4T
                    ;                                   .
 0045    C8         INTER1: RET Z       ;POUR 15T       . 5T
 0046    D1                 POP DE      ;ANNULE PILE    .10T
 0047    C8                 RET Z       ;RETARD 17 T    . 5T
 0048    18 FA              JR  INTER2  ;" ET AFFICHE   .12T
                    ;- - - - - - - - - - - - - - - - - -.

;AFFICHAGE SUR L'ECRAN

 004A    21 4030    DISPLA: LD  HL,FICAFF;DEB. AFFICHAGE.10T
 004D    CB FC              SET 7,H     ;POUR A15=0     . 8T
 004F    06 1A              LD  B,NLIGNE;26 LIGNES      . 7T
 0051    0E 08              LD  C,08    ;COMPTEUR MATRIC. 7T
 0053    D3 FF              OUT (0FF),A ;1er TOP SYNCHRO.11T
 0055    37                 SCF         ;POUR SYNCHRO   . 4T
 0056    D0                 RET NC      ;15 T EN 2 INSTR. 5T
 0057    D1                 POP DE      ;       "       .10T
 0058    D5                 PUSH DE     ;RETARD 27 T    .11T
 0059    00                 NOP         ;       "       . 4T
 005A    18 E8              JR  INTER2  ; " ET AFFICHE  .12T
                    ;- - - - - - - - - - - - - - - - - -.

;INTERRUPTION NMI

 0066    08         NMI:    EX  AF,AF'  ;Nbre DE LIGNES .
                            ;VIDES RESTANT  .
 0067    3D                 DEC A       ;-1             .
 0068    28 03              JR  Z,NMI001;DERNIERE ?     .
 006A    08                 EX  AF,AF'  ;NON            .
 006B    ED 45              RETN        ;FINI           .
                    ;                                   .
 006D    C5         NMI001: PUSH BC     ;SAUVE CONTEXTE .
 006E    D5                 PUSH DE     ;DU MICRO-      .
 006F    E5                 PUSH HL     ;PROCESSEUR     .
                            ;TEST SI HAUT OU.
 0070    FD CB 2B 5E        BIT 3,(IY+ECRIMP-DEBRAM);BAS.
 0074    28 1D              JR  Z,NMI002;SAUT SI FIN BAS.
 0076    3E 27              LD  A,NBVIDE;PREPARE POUR   .
 0078    08                 EX  AF,AF'  ;PROCHAIN       .
 0079    76                 HALT        ;ATTENTE NMI    .
                    ;                                   .
 007A    D3 FD              OUT (0FD),A ;STOPPE NMI     .
 007C    FD CB 2B 9E        RES 3,(IY+ECRIMP-DEBRAM);INI.
```

FIG.8a

PLANCHE 7/18

```
0080    F5              PUSH AF      ;SAUVE CONTEXTE .
0081    06 03           LD  B,3      ;RETARD 41 T    .
0083    10 FE    NMI111: DJNZ NMI111 ;       "       .
0085    CD 004A         CALL DISPLA ;AFFICHE PARTIE .
                                     ;UTILE          .
0088    D3 FE           OUT (0FE),A ;AUTORISE NMI   .
008A    CD 0124         CALL CLADIS ;GERE CLAVIER   .
008D    F1              POP AF      ;RESTITUTION    .
008E    E1              POP HL      ;DU CONTEXTE    .
008F    D1              POP DE      ;       "       .
0090    C1              POP BC      ;       "       .
0091    ED 45           RETN        ;FINI           .
                ;                                   .
0093    D3 FD    NMI002: OUT (0FD),A ;STOPPE LES NMI .
0095    3E 3E           LD  A,NHVIDE;PREPARE POUR   .
0097    08              EX  AF,AF'  ;PROCHAIN       .
0098    FD CB 2B DE     SET 3,(IY+ECRIMP-DEBRAM) ;  .
009C    F5              PUSH AF     ;SAUVEGARDE     .
                ;                                   .
                ;CLIGNOTEMENT DU CURSEUR             .
009D    2A 4019         LD  HL,(TEMCUR) ;TESTE SI LA.
00A0    7C              LD  A,H     ;TEMPO EST      .
00A1    B5              OR  L       ;TERMINEE       .
00A2    20 2A           JR  NZ,CURCL1 ;SAUT SI NON  .
00A4    21 000C         LD  HL,TEMDER ;REINITIALISE .
00A7    22 4019         LD  (TEMCUR),HL ;COMPTEUR   .
00AA    2A 4011         LD  HL,(CURSAF) ;COMPLEMEN- .
00AD    CB 16           RL  (HL)    ;TER BIT 7 DU   .
00AF    3F              CCF         ;CURSEUR        .
00B0    CB 1E           RR  (HL)    ;               .
                ;                                   .
00B2    06 1C    CURCL2: LD  B,1C    ;RETARD POUR    .
00B4    10 FE    NMI003: DJNZ NMI003 ;AJUSTER A 20 ms.
00B6    01 FEFE         LD  BC,0FEFE;GENERATION     .
00B9    ED 78    NMI005: IN  A,(C)   ;DES SIGNAUX DE .
00BB    CB 00           RLC B       ;SYNCHRO TRAME  .
00BD    38 FA           JR  C,NMI005;               .
00BF    2A 4029         LD  HL,(FRAMES);COMPTEUR DE-.
00C2    2B              DEC HL      ;CREMENTE TOUTE .
00C3    22 4029         LD  (FRAMES),HL;LES 20 ms   .
00C6    F1              POP AF      ;RESTITUTION    .
00C7    E1              POP HL      ;DU CONTEXTE    .
00C8    D1              POP DE      ;DU MICRO-      .
00C9    C1              POP BC      ;PROCESSEUR     .
00CA    D3 FE           OUT (0FE),A ;AUTORISE NMI   .
00CC    ED 45           RETN        ;FINI           .
                ;                                   .
                ;TEMPO NON TERMINEE POUR CLIGNOTEMENT
00CE    2B       CURCL1: DEC HL      ;CONTINUER      .
00CF    22 4019         LD  (TEMCUR),HL ;LA TEMPO   .
00D2    E5              PUSH        HL      ;RETARD 37 T
00D3    E1              POP HL      ;       "       .
00D4    22 4019         LD  (TEMCUR),HL ;   "       .
00D7    18 D9           JR  CURCL2  ;               .
```

FIG.8b

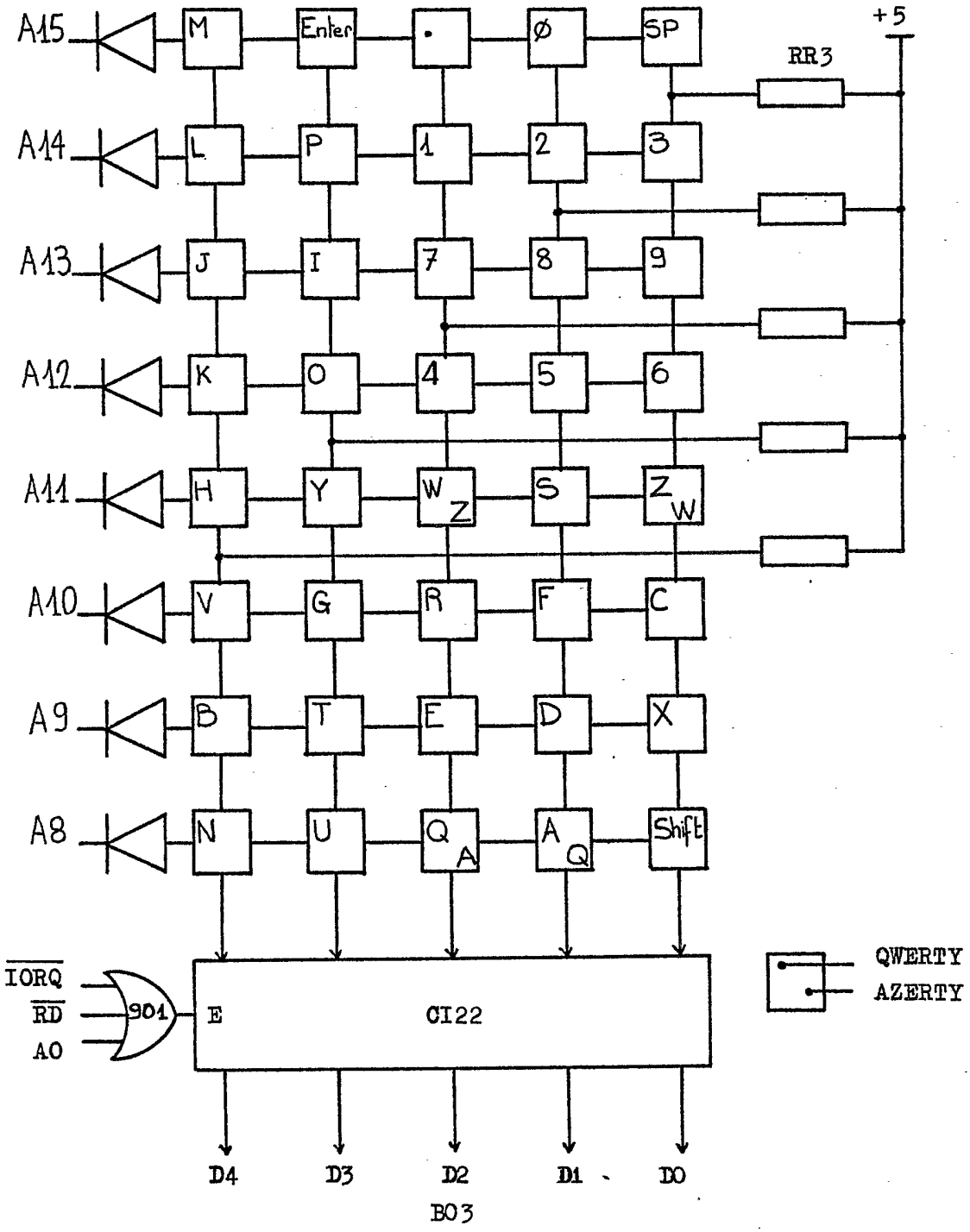

A15
A14
A13
A12
A11
A10
A9
A8
M
Enter
.
Ø
SP
L
P
1
2
3
J
I
7
8
9
K
O
4
5
6
H
Y
W
Z
S
Z
W
V
G
R
F
C
B
T
E
D
X
N
U
Q
A
A
Q
Shift
+5
RR3
IORQ
RD
A0
901
E
CI22
QWERTY
AZERTY
D4
D3
D2
D1
D0
BO3

FIG.9

```
                ;**************************************
                ;MODULE SCRCLA - SCRUTATION DU CLAVIER  .
                ;DONNE 2 OCTETS EN B ET C SIGNIFICATIFS .
                ;DE LA TOUCHE APPUYEE                  .

0142    16 FF   SCRCLA: LD    D,0FF     ;BIT 5 DE D=1 SI.
                                        ;SHIFT APPUYE   .
0144    01 FEFE         LD    BC,0FEFE  ;C=FE, ADRESSE  .
                                        ;DU CLAVIER     .
                                        ;B=FE, LIGNE 1  .
0147    ED 78   SCRCL2: IN    A,(C)     ;LECTURE        .
0149    F6 E0           OR    0E0       ;FORMAT 111xxxxx.
014B    FE FF   SCRCL4: CP    0FF       ;TOUCHE APPUYEE .
014D    20 07           JR    NZ,SCRCL1 ;OUI, SAUT      .
014F    CB 00           RLC   B         ;NON ESSAI LIGNE.
                                        ;SUIVANTE       .
0151    38 F4           JR    C,SCRCL2  ;DERNIERE LIGNE .
0153    A2      SCRCL3: AND   D         ;INDIC SHIFT O/N.
0154    4F              LD    C,A       ;RANGE L'INDICA-.
                                        ;TION DE COLONNE.
                                        ;B CONTIENT LA  .
                                        ;LIGNE          .
0155    C9              RET             ;FINI           .
                ;                                       .
0156    CB 40   SCRCL1: BIT   0,B       ;LIGNE DU SHIFT?.
0158    20 F9           JR    NZ,SCRCL3 ;SAUT SI NON    .
015A    CB 47           BIT   0,A       ;OUI. SHIFT ?   .
015C    20 F5           JR    NZ,SCRCL3 ;SAUT SI NON    .
015E    16 DF           LD    D,0DF     ;OUI BIT D5= 0  .
0160    F6 01           OR    01        ;SUPPRIME SHIFT .
0162    18 E7           JR    SCRCL4    ;AUTRE TOUCHE   .
                                        ;SUR LA LIGNE DU.
                                        ;SHIFT ?        .
```

FIG.10

PLANCHE IO/I8

TOUCHE APPUYEE	S	D	1			2			3			4		
QWERTY/ AZERTY	B C		a	b	c	a	b	c	a	b	c	a	b	c
Shift	FEDF	00			1			2			3			4
Enter	7FF7	1F	Enter	0D	1	Ent.	0D	1			.			2
Espace	7FFE	07	Espace	20	1	Esp	20	2			.		A0	4
.	7FFB	17	.	2E	1	.	2E	2	NEW	8DAE	2	.	AE	4
0	7FFD	0F	0	30	1	0	30	2	MACH.	8DB0	2	0	B0	4
1	BFFB	16	1	31	1	1	31	2	RAPIDE	8DB1	.	1	B1	4
2	BFFD	0E	2	32	1	2	32	2			.	2	B2	4
3	BFFE	06	3	33	1	3	33	2	NORMAL	8DB3	.	3	B3	4
4	EFFB	14	4	34	1	4	34	2			.	4	B4	4
5	EFFD	0C	5	35	1	5	35	2	GRAPH.	8DB5	4	5	B5	4
6	EFFE	04	6	36	1	6	36	2			.	6	B6	4
7	DFFB	15	7	37	1	7	37	2	MAJmin	8DB7	.	7	B7	4
8	DFFD	0D	8	38	1	8	38	2			.	8	B8	4
9	DFFE	05	9	39	1	9	39	2	VIDEO	8DB9	2	9	B9	4
A/Q	FEFD	08	POKE	8D41	2	A a	41 61	2	ABS	8DC1	2	A a	C1 E1	4
B	FDEF	21	SAVE	8D42	2	B b	42 62	2	ASIN	8DC2	2	B b	C2 E2	4
C	FBFE	02	CONT	8D43	2	C c	43 63	2	COS	8DC3	2	C c	C3 E3	4
D	FDFD	09	DATA	8D44	2	D d	44 64	2	SQR	8DC4	2	D d	C4 E4	4
E	FDFB	11	REM	8D45	2	E e	45 65	2	NOT	8DC5	2	E e	C5 E5	4
F	FBFD	0A	FOR	8D46	2	F f	46 66	2	EXP	8DC6	2	F f	C6 E6	4
G	FBF7	1A	GOTO	8D47	2	G g	47 67	2	LN	8DC7	2	G g	C7 E7	4
H	F7EF	23	GOSUB	8D48	2	H h	48 68	2	INT	8DC8	2	H h	C8 E8	4
I	DFF7	1D	INPUT	8D49	2	I i	49 69	2	INKEY$	8DC9	2	I i	C9 E9	4
J	DFEF	25	LOAD	8D4A	2	J j	4A 6A	2	VAL	8DCA	2	J j	CA EA	4
K	EFEF	24	LET	8D4B	2	K k	4B 6B	2	LEN	8DCB	2	K k	CB EB	4
L	BFEF	26	LIST	8D4C	2	L l	4C 6C	2	LLIST	8DCC	2	L l	CC EC	4
M	7FEF	27	PAUSE	8D4D	2	M m	4D 6D	2	ATAN	8DCD	2	M m	CD ED	4
N	FEEF	20	NEXT	8D4E	2	N n	4E 6E	2	ACOS	8DCE	2	N n	CE EE	4
O	EFF7	1C	IF	8D4F	2	O o	4F 6F	2	COPY	8DCF	2	O o	CF EF	4
P	BFF7	1E	PRINT	8D50	2	P p	50 70	2	LPRINT	8DD0	2	P p	D0 F0	4
Q/A	FEFB	10	RAND	8D51	2	Q q	51 71	2	RND	8DD1	2	R q	D1 F1	4
R	FBFB	12	RESTORE	8D52	2	R r	52 72	2	OR	8DD2	2	R r	D2 F2	4
S	F7FD	0B	STOP	8D53	2	S s	53 73	2	SGN	8DD3	2	S s	D3 F3	4
T	FDF7	19	RETURN	8D54	2	T t	54 74	2	STR$	8DD4	2	T t	D4 F4	4
U	FEF7	18	UNPLOT	8D55	2	U u	55 75	2	USR	8DD5	2	U u	D5 F5	4
V	FBEF	22	VERIFY	8D56	2	V v	56 76	2	TAN	8DD6	2	V v	D6 F6	4
W/Z	F7FB	13	READ	8D57	2	W w	57 77	2	AND	8DD7	2	W w	D7 F7	4
X	FDFE	01	CLS	8D58	2	X x	58 78	2	SIN	8DD8	2	X x	D8 F8	4
Y	F7F7	1B	RUN	8D59	2	Y y	59 79	2	CHR$	8DD9	2	Y y	D9 F9	4
Z/W	F7FE	03	CLEAR	8D5A	2	Z z	5A 7A	2	PI	8DDA	2	Z z	DA FA	4

FIG. 11a

PLANCHE II/I8

TOUCHE APPUYEE	S	D	1			2			3			4		
QWERTY/ AZERTY	B C		a	b	c	a	b	c	a	b	c	a	b	c
Shift Ent	7FD7	47	Fonct.		3	Identique		3	Identique					2
Shift Esp	7FDE	2F	Espace	20	1	aux cas		2	au cas sans				08	4
Shift .	7FDB	3F	=	3D	1	1-a	1-b	2	Shift				17	4
Shift 0	7FDD	37	**	8D23	2			2	représenté				23	4
Shift 1	BFDB	3E	+	2B	1			2	dans la				16	4
Shift 2	BFDD	36	↓	8D21	1			2	colonne				21	4
Shift 3	BFDE	2E	-	2D	1			2	supérieure				5E	4
Shift 4	EFDB	3C	←	8D18									18	4
Shift 5	EFDD	34	Efface	7F								Eff	7F	4
Shift 6	EFDE	2C	→	8D5F									5F	4
Shift 7	DFDB	3D	*	2A	1			2					15	4
Shift 8	DFDD	35	↑	8D09	1			2					09	4
Shift 9	DFDE	2D	/	2F	1			2					5D	4
Shift A/Q	FEDD	30	PEEK	8D19	2			2					19	4
Shift B	FDCF	49	>=	8D5C	2			2					5C	4
Shift C	FBDE	2A	<=	8D05	2			2					05	4
Shift D	FDDD	31	DIM	8D1A	2			2					1A	4
Shift E	FDDB	39	EDIT	8D01	1			1					01	4
Shift F	FBDD	32	TO	8D1B	2			2					1B	4
Shift G	FBD7	42	STEP	8D26	2			2					26	4
Shift H	F7CF	4B	?	3F	1			2					25	4
Shift I	DFD7	45	(	28	1			2					5B	4
Shift J	DFCF	4D	:	3A	1			2					10	4
Shift K	EFCF	4C	,	2C	1			2					27	4
Shift L	BFCF	4E	;	3B	1			2					0B	4
Shift M	7FCF	4F	>	3E	1			2					04	4
Shift N	FECF	48	<	3C	1			2					07	4
Shift O	EFD7	44	)	29	1			2					1E	4
Shift P	BFD7	46	"	22	1			2					1D	4
Shift Q/A	FEDB	38	$	24	1			2					00	4
Shift R	FBDB	3A	AT	8D02	2			2					02	4
Shift S	F7DD	33	SCROLL	8D1C	2			2					1C	4
Shift T	FDD7	41	TAB	8D7E	2			2					7E	4
Shift U	FED7	40	PLOT	8D7D	2			2					7D	4
Shift V	FBCF	4A	<>	8D06	2			2					06	4
Shift W/Z	F7DB	3B	RENUMB	8D03	2			2					03	4
Shift X	FDDE	29	""	8D7C	2			2					7C	4
Shift Y	F7D7	43	THEN	8D1F	1			1					1F	4
Shift Z/W	F7DE	2B	CODE	8D7B	2			2					7B	4
Aucune	FEFF	00			1			2			3			4

FIG.11b

PLANCHE I2/I8

```
                ;*************************************
                ;MODULE DECCLA                       .
                ;A PARTIR DES REGISTRES B ET C FOURNIS .
                ;PAR SCRCLA, GENERE EN B UN OCTET SIGNI-.
                ;FICATIF DE LA TOUCHE APPUYEE. CET OCTET.
                ;PERMETTRA A CARMOD DE TROUVER DANS UNE .
                ;TABLE CONTINUE LA VALEUR DE LA TOUCHE  .

0164    16 00   DECCLA: LD      D,0       ;PAS DE SHIFT    .
0166    CB 69           BIT     5,C       ;SHIFT ?         .
0168    20 04           JR      NZ,DECCL1 ;SAUT SI NON     .
016A    16 28           LD      D,28      ;OUI D=28        .
016C    CB E9           SET     5,C       ;SUPPRIME SHIFT  .
016E    AF      DECCL1: XOR     A         ;INIT. A=0       .
016F    CB 09   DECCL3: RRC     C         ;POUR TEST NUME-.
0171    30 08           JR      NC,DECCL2 ;RO DE COLONNE   .
0173    3C              INC     A         ;NON A=A+1       .
0174    FE 05           CP      05        ;COLONNE 5 ?     .
0176    20 F7           JR      NZ,DECCL3 ;SINON SUIVANTE.
0178    AF      DECCL6: XOR     A         ;OUI, A=0        .
0179    47      DECCL7: LD      B,A       ;SAUVEGARDE EN B.
017A    C9              RET               ;FINI            .
                ;                                          .
017B    5F      DECCL2: LD      E,A       ;COLONNE TROUVEE.
                                          ;RECHERCHE LIGNE.
017C    3E FF           LD      A,0FF     ;INIT COMPTEUR   .
017E    3C      DECCL8: INC     A         ;COMPTE          .
017F    CB 08           RRC     B         ;TESTE LIGNE     .
0181    30 FB           JR      NC,DECCL8 ;TROUVEE ?       .
0183    CB 03           RLC     E         ;MISE AU FORMAT  .
0185    CB 03           RLC     E         ;E=0000 0XXX     .
0187    CB 03           RLC     E         ;A=0000 0YYY     .
0189    83              ADD     A,E       ;FORMAT 00XXXYYY.
018A    82              ADD     A,D       ;PRISE DE SHIFT  .
018B    18 EC           JR      DECCL7    ;                .
```

FIG. 12

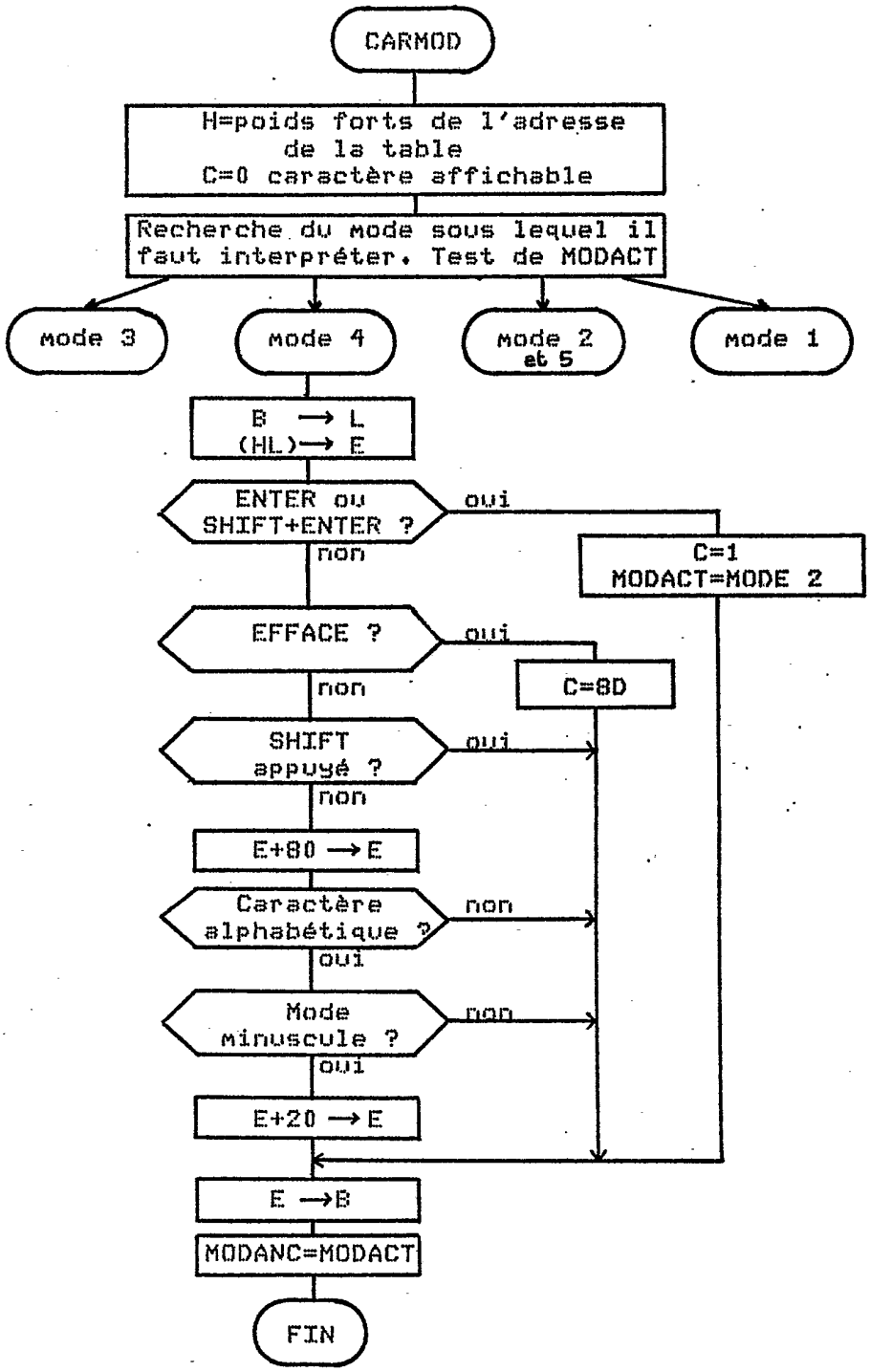

CARMOD
H=poids forts de l'adresse de la table
C=0 caractère affichable
Recherche du mode sous lequel il faut interpréter. Test de MODACT
mode 3
mode 4
mode 2 et 5
mode 1
B → L
(HL) → E
ENTER ou SHIFT+ENTER ?
oui
non
C=1
MODACT=MODE 2
EFFACE ?
oui
non
C=8D
SHIFT appuyé ?
oui
non
E+80 → E
Caractère alphabétique ?
non
oui
Mode minuscule ?
non
oui
E+20 → E
E → B
MODANC=MODACT
FIN

FIG. 13a

PLANCHE I4/I8

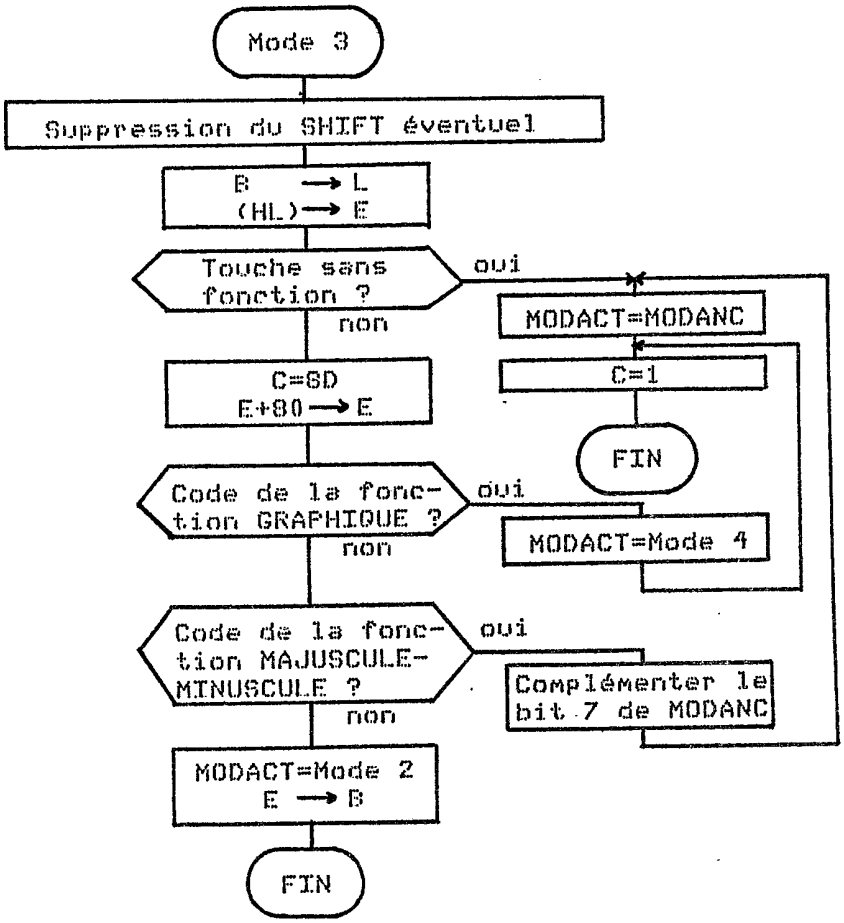


FIG.13b

PLANCHE I5/I8

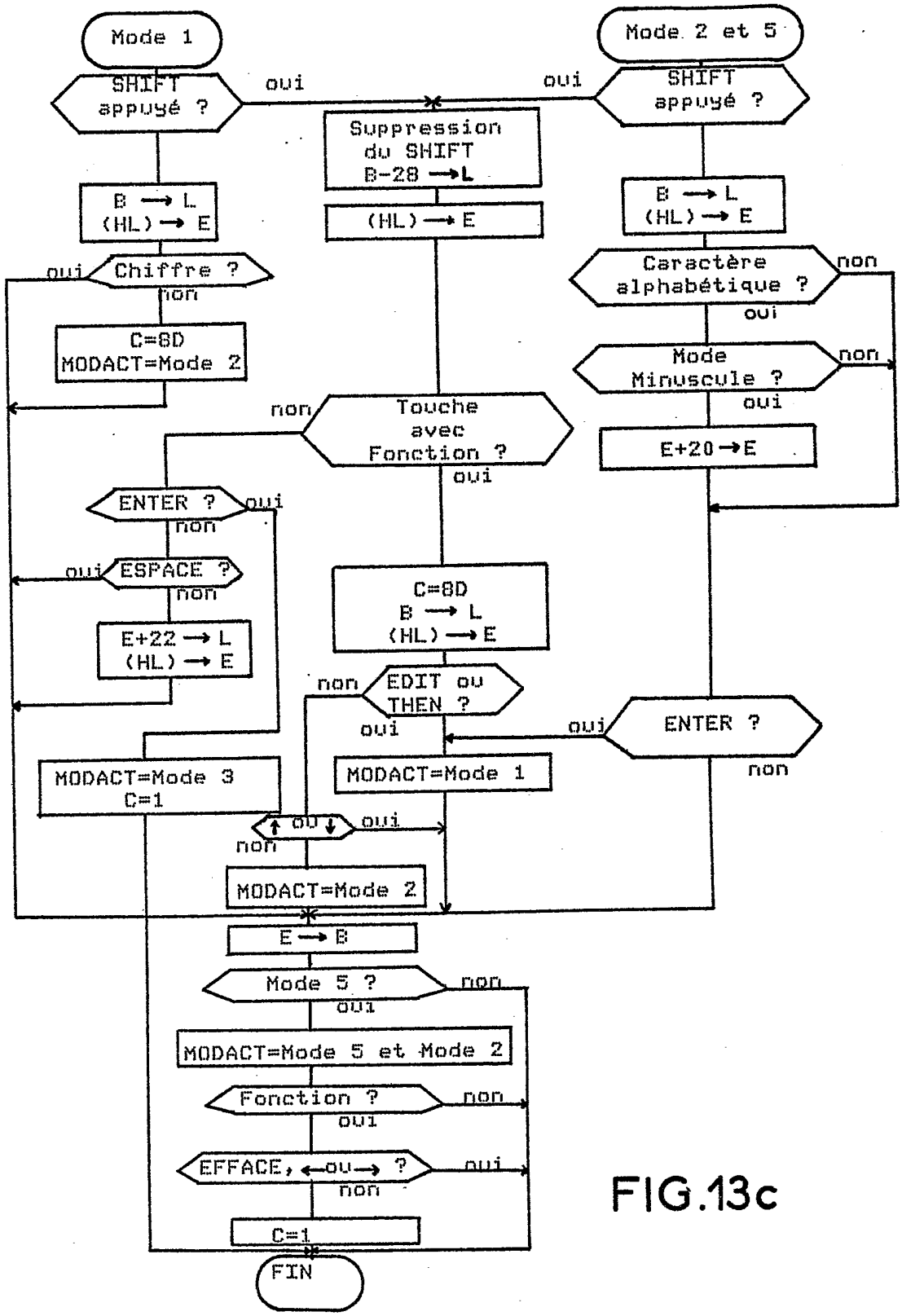


FIG.13c

PLANCHE I6/I8

```
TCASCI          ;**********************************
                ;TABLEAU DONNANT LE CODE ASCII DE LA   .
                ;TOUCHE APPUYEE EN FONCTION DE L'OCTET .
                ;FOURNI PAR DECCLA. LE CODE FOURNI SERA.
                ;INTERPRETE SELON LE MODE PAR CARMOD   .
```

ADR.	CODE	CAR.
1B01	58	X
1B02	43	C
1B03	5A/57	Z/W
1B04	36	6
1B05	39	9
1B06	33	3
1B07	20	SPACE
1B08	41/51	A/Q
1B09	44	D
1B0A	46	F
1B0B	53	S
1B0C	35	5
1B0D	38	8
1B0E	32	2
1B0F	30	0
1B10	51/41	Q/A
1B11	45	E
1B12	52	R
1B13	57/5A	W/Z
1B14	34	4
1B15	37	7
1B16	31	1
1B17	2E	.
1B18	55	U
1B19	54	T
1B1A	47	G
1B1B	59	Y
1B1C	4F	O
1B1D	49	I
1B1E	50	P
1B1F	0D	ENTER
1B20	4E	N
1B21	42	B
1B22	56	V
1B23	48	H
1B24	4B	K
1B25	4A	J
1B26	4C	L
1B27	4D	M
1B28	00	
1B29	7C	GR X
1B2A	05	GR C
1B2B	7B	GR Z
1B2C	5F	GR 6
1B2D	5D	GR 9
1B2E	5E	GR 3
1B2F	08	GR SPACE
1B30	19	GR A
1B31	1A	GR D
1B32	1B	GR F
1B33	1C	GR S
1B34	7F	DELETE
1B35	09	GR 8
1B36	21	GR 2
1B37	23	GR 0
1B38	00	GR Q
1B39	01	GR E
1B3A	02	GR R
1B3B	03	GR W
1B3C	18	GR 4
1B3D	15	GR 7
1B3E	16	GR 1
1B3F	17	GR .
1B40	7D	GR U
1B41	7E	GR T
1B42	26	GR G
1B43	1F	GR Y
1B44	1E	GR O
1B45	5B	GR I
1B46	1D	GR P
1B47	8C	SHIFT ENTER
1B48	07	GR N
1B49	5C	GR B
1B4A	06	GR V
1B4B	25	GR H
1B4C	27	GR K
1B4D	10	GR J
1B4E	0B	GR L
1B4F	04	GR M
1B50	3D	=
1B51	00	
1B52	00	
1B53	2B	+
1B54	00	
1B55	2D	-
1B59	2A	*
1B5A	00	
1B5B	2F	/
1B6A	3F	?
1B6B	28	(
1B6C	3A	:
1B6D	2C	,
1B6E	3B	;
1B6F	3E	>
1B70	3C	<
1B71	29	)
1B72	22	"
1B73	24	$

FIG.14

PLANCHE I7/I8

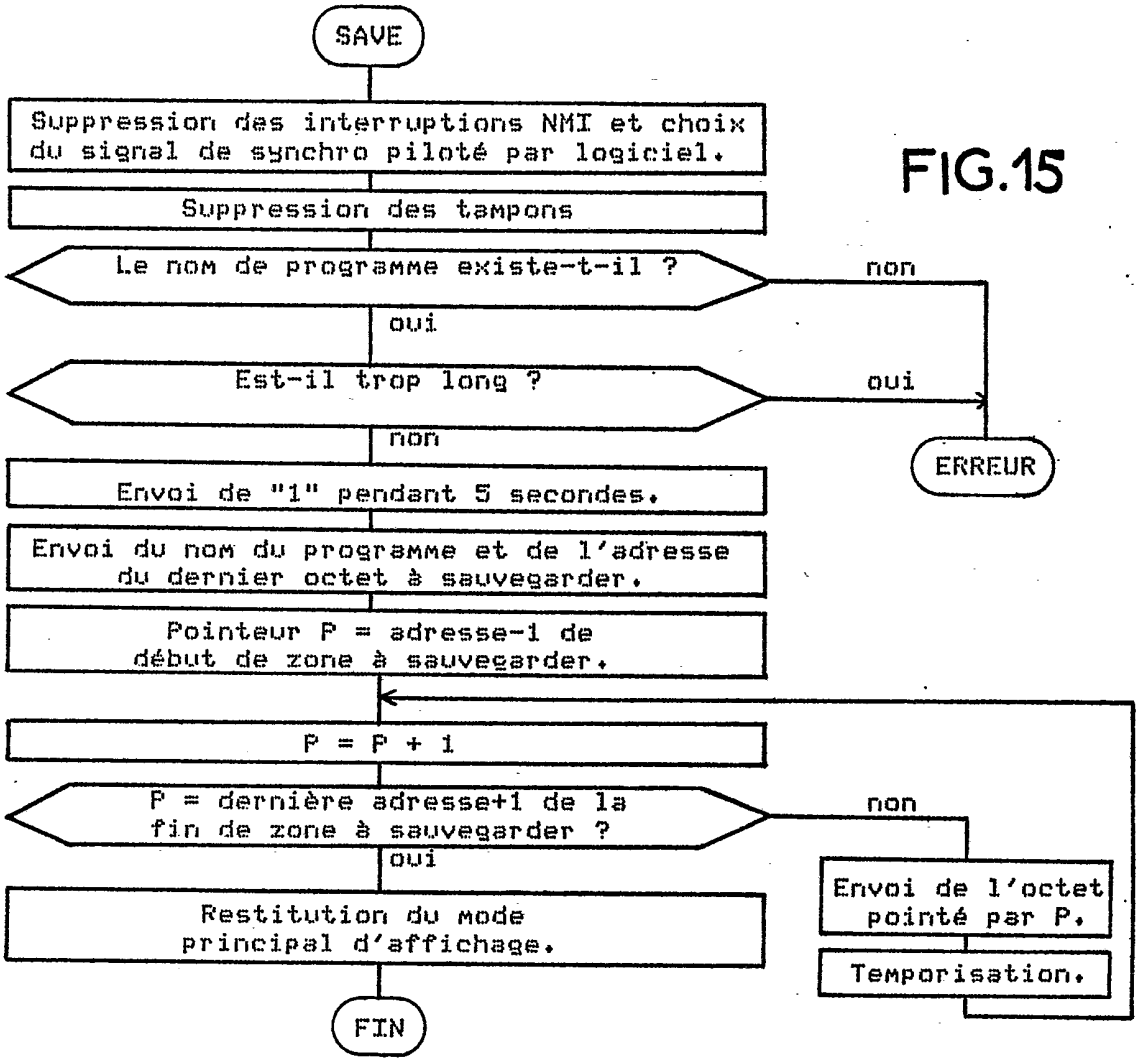


FIG.15

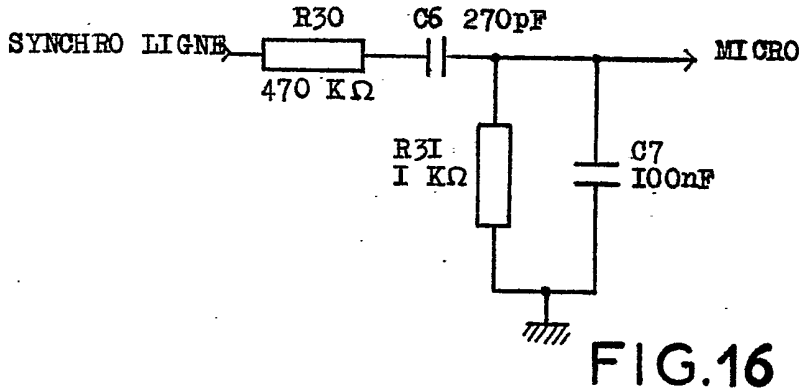


FIG.16

PLANCHE I8/I8

FIG.17

```
;ENVOI D'UN BIT 0

2053 C5          EZERO:  PUSH   BC       ;                    .
2054 0E 04               LD     C,04     ;4 TOPS              .
2056 10 FE       EZERO1: DJNZ   EZERO1   ;TEMPO               .
2058 3E 7F               LD     A,7F     ;SYNCHRO=1           .
205A DB FE               IN     A,(0FE)  ;                    .
205C 1F                  RRA             ;                    .
205D D2 080F             JP     NC,ARRE16;BREAK APPUYE        .
2060 06 67               LD     B,67     ;POUR TEMPO          .
2062 10 FE       EZERO2: DJNZ   EZERO2   ;TEMPO               .
2064 D3 FF               OUT    (0FF),A  ;SYNCHRO=0           .
2066 0D                  DEC    C        ;COMPTE LES TOPS     .
2067 28 06               JR     Z,EZERO4 ;SAUT SI DERNIER     .
2069 C8                  RET    Z        ;AJUSTAGE 10 T       .
206A C8                  RET    Z        ;"                   .
206B 06 65               LD     B,65     ;POUR TEMPO          .
206D 18 E7               JR     EZERO1   ;TOP SUIVANT         .
                         ;                                    .
206F 00          EZERO4: NOP              ;RETARD 4 T          .
2070 06 35               LD     B,35     ;                    .
2072 10 FE       EZERO5: DJNZ   EZERO5   ;TEMPO               .
2074 C1          EUN4:   POP    BC       ;                    .
2075 C9                  RET             ;                    .
;.....................................................
```

FIG.18

```
;ENVOI D'UN BIT 1

2076     C5        EUN:    PUSH   BC       ;                 .
2077     0E 08              LD     C,08     ;8 TOPS           .
2079     10 FE     EUN1:   DJNZ   EUN1     ;TEMPO            .
207B     3E 7F              LD     A,7F     ;SYNCHRO=1        .
207D     DB FE              IN     A,(0FE)  ;                 .
207F     1F                 RRA             ;                 .
2080     D2 080F   EUN5:   JP     NC,ARRE16;BREAK APPUYE     .
2083     FD 86 00           ADD    A,(IY+0);RETARD 19 T       .
2086     06 31              LD     B,31     ;                 .
2088     10 FE     EUN3:   DJNZ   EUN3     ;TEMPO            .
208A     D3 FF              OUT    (0FF),A  ;SYNCHRO=0        .
208C     0D                 DEC    C        ;COMPTE LES TOPS  .
208D     28 E5              JR     Z,EUN4   ;SAUT SI DERNIER  .
208F     00                 NOP             ;AJUSTAGE 16 T    .
2090     03                 INC    BC       ; "               .
2091     0B                 DEC    BC       ; "               .
2092     06 30              LD     B,30     ;POUR CALAGE      .
2094     18 E3              JR     EUN1     ;                 .
```

FIG.19

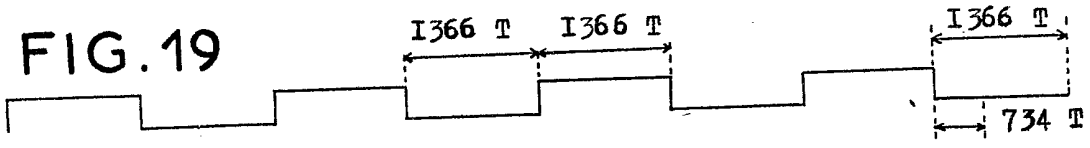


FIG.20

