(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 591 006**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **86 16559**

(51) Int Cl⁴ : G 06 K 19/00; G 06 F 12/02.

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt** : 27 novembre 1986.

(30) **Priorité** : JP, 30 novembre 1985, n° 269629/85; 4 septembre 1986, n° 208252/86.

(43) **Date de la mise à disposition du public de la demande** : BOPI « Brevets » n° 23 du 5 juin 1987.

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

(71) **Demandeur(s)** : *Société dite : KABUSHIKI KAISHA TOSHIBA.* — JP.

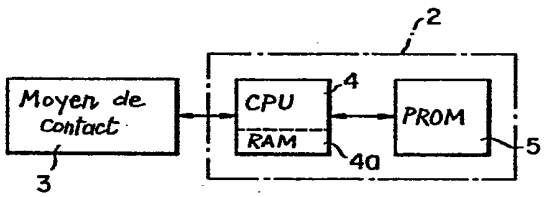
(72) **Inventeur(s)** : Yasuo Iijima.

(73) **Titulaire(s)** :

(74) **Mandataire(s)** : Cabinet Beau de Loménie.

(54) Dispositif électronique portatif.

(57) Selon le dispositif électronique portatif de l'invention, lorsqu'une donnée est écrite dans une PROM 5, l'adresse de début de la zone non écrite suivante est emmagasinée dans une RAM 4*a* incorporée dans une CPU 4. Une mémoire de données est divisée en plusieurs zones, et chaque zone est en outre divisée en plusieurs blocs. La longueur de donnée de chaque bloc est déterminée pour chaque zone, et la donnée d'entrée est divisée en fonction de la longueur de donnée comme une unité de traitement en vue de son emmagasinage. Après l'emmagasinage du bloc final, la CPU 4 emmagasine la position d'adresse de la donnée finale du bloc final au début de la zone correspondante.



FR 2 591 006 A1

La présente invention concerne un dispositif électronique portatif, ce que l'on appelle une carte à circuit intégré, qui contient une puce IC (à circuit intégré) possédant une mémoire de données permanente, et un élément de commande, par exemple une unité centrale de traitement (CPU).

Ces dernières années, il a été mis au point, comme carte de mémoire servant à emmagasiner les données, une carte à circuit intégré incorporant une puce IC qui possède une mémoire permanente (par exemple une mémoire fixe programmable, ou PROM) et une CPU, cette carte à circuit intégré étant disponible dans le commerce. Pour un système utilisant une carte à circuit intégré de ce type, les échanges de données avec la carte à circuit intégré s'effectuent à l'aide d'un lecteur-enregistreur de cartes. Lorsque la carte à circuit intégré reçoit une donnée d'instruction à l'aide d'un code de fonction en provenance du lecteur-enregistreur de cartes, celle-ci exécute une fonction et délivre le résultat de l'exécution au lecteur-enregistreur de cartes au titre de donnée de réponse.

Lorsqu'un signal d'entrée se présentant sous la forme d'une chaîne de données qui a été appliquée à la carte à circuit intégré s'inscrit dans la mémoire de données, la donnée d'entrée ne peut être écrite qu'après qu'une zone non écrite, dans laquelle aucune donnée valable n'est écrite, a été certifiée. Dans ce cas, lorsque plusieurs chaînes de données sont successivement écrites, si la donnée d'adresse de la zone suivante (par exemple une adresse de début de zone non écrite) dans laquelle la donnée doit être écrite n'est pas emmagasinée, une zone non écrite doit d'abord être certifiée à chaque fois que la chaîne de données est écrite, ce qui conduit à une longue durée totale d'écriture.

Dans une carte à circuit intégré classique, lorsqu'une chaîne de données s'inscrit dans une zone cible, elle s'inscrit à partir d'une adresse de référence donnée (par exemple une adresse de début) suivant un seul sens, et une adresse finale pour la chaîne de données écrite est emmagasinée dans une mémoire vive (RAM) d'un élément de commande, afin de simplifier l'opération d'écriture suivante. Plus spécialement, lorsque l'on met en service l'alimentation électrique de la carte à circuit intégré, il est possible

de réaliser la simplification ci-dessus indiquée. Toutefois, après que l'alimentation électrique a été temporairement éteinte et est de nouveau remise en service afin de permettre la réalisation d'une opération d'écriture, le contenu de la RAM a été effacé. Par conséquent, il faut chercher l'adresse finale à partir de la zone cible. Dans ce cas, une valeur spécifique, par exemple "FF_H", doit être maintenue comme donnée permettant de certifier une donnée non écrite. Pour cette raison, la vitesse d'écriture de données diminue considérablement.

C'est un but de l'invention de proposer un dispositif électronique portatif qui peut effectuer en continu une opération d'écriture de données dans une mémoire de données sans devoir séquentiellement certifier des zones non écrites et qui peut améliorer la vitesse d'écriture et abréger le temps d'écriture.

Un autre but de l'invention est de proposer un dispositif électronique portatif qui peut fortement améliorer la vitesse d'accès de données et qui ne nécessite l'existence d'aucune donnée permettant de certifier des données non écrites.

Pour atteindre le but ci-dessus indiqué, il est proposé un dispositif électronique portatif comprenant :

un moyen de mémoire servant à emmagasiner des données ; et

une unité centrale de traitement servant à commander sélectivement l'accès pour lecture de données ou l'accès pour écriture de données dans le moyen de mémoire en réponse à une instruction délivrée par un système principal connecté au dispositif électronique portatif, l'unité centrale de traitement emmagasinant des données, qui peuvent fournir l'adresse d'emmagasinage de la zone non écrite suivante, dans une zone prédéterminée du moyen de mémoire lorsque l'unité centrale de traitement écrit des données dans le moyen de mémoire.

La description suivante, conçue à titre d'illustration de l'invention, vise à donner une meilleure compréhension de ses caractéristiques et avantages ; elle s'appuie sur les dessins annexés, parmi lesquels :

la figure 1 est une vue en perspective montrant l'aspect extérieur d'une carte à circuit intégré selon un mode de réalisation de l'invention ;

la figure 2 est un schéma de principe montrant la disposition d'une puce à circuit intégré contenue dans la carte à circuit intégré ;

la figure 3 est un schéma de principe montrant la disposition d'un manipulateur de cartes ;

la figure 4 est un organigramme servant à expliquer une opération d'écriture dans la carte à circuit intégré ;

la figure 5 montre le format d'une donnée d'instruction d'écriture ;

la figure 6 est un organigramme permettant d'expliquer une opération de certification de zone non écrite pour la carte à circuit intégré ;

la figure 7 montre le format d'une donnée d'instruction de certification de zone non écrite ;

la figure 8 est un schéma de principe de fonction d'une carte à circuit intégré selon un autre mode de réalisation de l'invention ;

la figure 9 est un schéma de principe montrant la disposition d'une puce à circuit intégré contenue dans la carte à circuit intégré que représente la figure 8 ;

la figure 10 est une carte de mémorisation pour une mémoire de données divisée en plusieurs blocs ;

la figure 11 est un exemple de table d'index montrant la relation qui existe entre les numéros des zones, les nombres de bytes, les adresses de début et les adresses finales de zones respectives de la mémoire de données ;

la figure 12 montre le format d'une donnée d'attribut ;

les figures 13A et 13B sont des organigrammes permettant d'expliquer une opération d'écriture de données selon un autre mode de réalisation de l'invention ;

la figure 14 montre le format d'une donnée d'instruction d'écriture ;

les figures 15A et 15B sont des organigrammes permettant d'expliquer une opération d'écriture en continuation (à suivre) ;

la figure 16 montre le format d'une donnée d'instruction d'écriture en continuation ;

les figures 17A à 17J sont des diagrammes servant à expliquer en détail l'opération d'écriture de données ;

les figures 18A à 18C sont des organigrammes permettant d'expliquer une opération de lecture de données ;

la figure 19 montre le format d'une donnée d'instruction de lecture ;

les figures 20A et 20B sont des organigrammes permettant d'expliquer une opération de lecture en continuation ;

la figure 20C est une vue montrant un exemple du format d'une donnée d'instruction de lecture en continuation ;

les figures 21A à 21F sont des diagrammes permettant d'expliquer en détail l'opération de lecture de données ;

la figure 22 est un organigramme permettant d'expliquer une opération d'effacement de données ; et

la figure 23 montre le format d'une donnée d'instruction d'effacement.

Sur la figure 1, est représentée une carte à circuit intégré 1 constituant un dispositif électronique portatif selon l'invention. Comme le montre la figure 2, la carte à circuit intégré 1 contient une puce à circuit intégré 2, laquelle est électriquement connectée à un moyen de contact 3 constitué de plusieurs contacts formés à la surface de la carte à circuit intégré 1. La puce à circuit intégré 2 comprend une unité centale de traitement (CPU) 4 faisant fonction d'élément de commande pour le déchiffrage, le traitement et l'emmagasinage d'instructions et de données d'entrée, et une mémoire morte programmable (PROM) 5, faisant fonction de mémoire de données permanente, qui est soumise à des opérations de lecture-écriture sous commande de la CPU 4. On note que la CPU 4 contient une mémoire vive 4a.

La figure 3 montre la disposition d'un manipulateur de cartes utilisé comme dispositif terminal d'un réseau bancaire ou d'un réseau d'achat à domicile, auquel on applique la carte à circuit intégré 1 comme dispositif électronique portatif de

l'invention. Le manipulateur permet de connecter la carte à circuit intégré 1 à un dispositif de commande (par exemple CPU) 12 via un lecteur-enregistreur de cartes 11. La CPU 12 est connectée à un clavier 13, un dispositif d'affichage cathodique 14, une imprimante 15, et une unité de traitement de disques souples 16.

La carte à circuit intégré 1 déchiffre la donnée d'instruction d'entrée venant du lecteur-enregistreur de cartes 11 et exécute le traitement nécessaire. Pour réaliser cela, la carte à circuit intégré 1 est dotée d'une CPU 4 et d'une PROM 5, comme ci-dessus indiqué. Le lecteur-enregistreur de cartes 11 assure la transmission des données, par exemple données d'instruction et données de réponse, entre la carte à circuit intégré 1 et la CPU 12.

On va maintenant décrire le fonctionnement de la carte à circuit 1 ayant la disposition ci-dessus indiquée. Tout d'abord, on va décrire une opération d'écriture en relation avec l'organigramme de la figure 4. La carte à circuit intégré 1 est en attente d'une donnée d'instruction provenant du lecteur-enregistreur de cartes 11. La donnée d'instruction d'écriture est constituée par exemple d'un code de fonction d'instruction d'écriture et de données d'emmagasinage, comme représenté sur la figure 5. S'il est déterminé, à l'étape 21, que la donnée d'instruction d'écriture venant du lecteur-enregistreur de cartes 11 a été reçue, la CPU 4 vérifie, à l'étape 23, si l'adresse de début de zone non écrite est emmagasinée dans une zone prédéterminée de sa RAM interne 4a. Si la réponse est NON à l'étape 23, la CPU 4 délivre une donnée de réponse indiquant la non-existence de l'adresse de départ pour la zone non écrite dans l'étape 37, et le programme revient à l'étape 21. Toutefois, si la réponse est OUI à l'étape 23, la CPU 4 écrit les données d'emmagasinage se trouvant dans la donnée d'instruction dans la zone de la PROM 5 indiquée par l'adresse de début, et emmagasine l'adresse de début dans la zone non écrite après l'opération d'écriture dans la zone prédéterminée de la RAM 4a afin de la remettre à jour au cours de l'étape 25. S'il est déterminé, à l'étape 27, qu'aucune zone non écrite n'existe dans la PROM 5, la CPU 4 emmagasine une adresse non assignée à des zones de données dans la RAM 4a au cours de l'étape 33, et elle délivre une donnée de réponse indiquant

l'achèvement de l'écriture et la non-existence d'une adresse d'écriture suivante au cours de l'étape 35. Le programme revient alors à l'étape 21. Toutefois, s'il est déterminé, à l'étape 27, que la nouvelle zone d'écriture existe dans la PROM 5, la CPU 4 emmagasine l'adresse de début de la nouvelle zone d'écriture dans la zone prédéterminée de la RAM 4a au cours de l'étape 29. A l'étape 31, la CPU 4 délivre une donnée de réponse indiquant l'achèvement de l'écriture et l'existence de la zone d'écriture suivante, puis le programme revient à l'étape 21.

On va maintenant décrire, en relation avec l'organigramme de la figure 6, une opération visant à certifier l'adresse de début de la zone non écrite. La carte à circuit intégré 1 est d'abord en attente d'une donnée d'instruction certifiant la zone non écrite. La donnée d'instruction certifiant la zone non écrite comporte par exemple un code de fonction certifiant la zone non écrite, comme représenté sur la figure 7. Lorsqu'il est décidé, à l'étape 39, que la donnée d'instruction certifiant la zone non écrite à été reçue, la CPU 4 recherche l'adresse de début d'une zone non écrite lors de l'étape 41. Si la réponse est NON à l'étape 43, la CPU 4 délivre une donnée de réponse indiquant la non-existence de la zone non écrite au cours de l'étape 49, puis le programme revient à l'étape 39. Toutefois, si la réponse est OUI à l'étape 43, la CPU 4 emmagasine l'adresse de début dans la zone prédéterminée de la RAM 4a au cours de l'étape 45, et délivre une donnée de réponse indiquant l'existence de la zone non écrite au cours de l'étape 47. Le programme revient ensuite à l'étape 39.

Lorsque la donnée a été écrite dans la carte à circuit intégré 1, après que la donnée d'instruction certifiant la zone non écrite a été reçue, une ou plusieurs données d'instruction d'écriture sont reçues.

Avec la disposition ci-dessus présentée, lorsque la donnée est inscrite dans la mémoire de données, l'adresse de début d'une zone non écrite immédiatement suivante est toujours emmagasinée. Ainsi, il n'est pas besoin de rechercher une zone non écrite à chaque fois qu'une donnée s'inscrit dans la mémoire de données,

et l'opération d'écriture de données peut s'exécuter en continu, ce qui améliore la vitesse d'écriture et abrège le temps d'écriture. Avant que la donnée ne soit écrite dans la mémoire de donnée, on certifie la zone non écrite pour emmagasiner son adresse de début, ce qui simplifie la séquence d'écriture de données suivante et raccourcit le temps d'écriture.

On va maintenant présenter un autre mode de réalisation de l'invention en relation avec les figures 8 à 23.

Comme le montre la figure 8, une carte à circuit intégré selon un autre mode de réalisation de l'invention comprend une section destinée à exécuter des fonctions de base, c'est-à-dire une section de lecture-écriture 51, une section de fixation-recueil de numéro d'identification personnel (PIN) 53, et une section de chiffrage-déchiffrage 55, ainsi qu'une superviseur 57 assurant la gestion des fonctions de base. La section de lecture-écriture 51 lit, écrit ou efface des données en relation avec le lecteur-enregistreur de cartes 11. La section de fixation-recueil de PIN 53 exécute l'opération consistant à emmagasiner, tout en en empêchant la lecture, un numéro d'identification personnel, ou PIN, fixé par l'utilisateur, et recueille sélectivement un PIN après que le PIN a été fixé pour fournir une autorisation de poursuite du traitement. La section de chiffrage-déchiffrage 55 chiffre la donnée ou déchiffre la donnée chiffrée de manière à empêcher que les données de transmission ne puissent être décrites ou contrefaites pendant leur transmission de la CPU 12 à un autre dispositif terminal via une ligne de transmission. La section 55 exécute un traitement de données en fonction d'un algorithme de chiffrage, par exemple un DES (Data Encryption Standard, ou norme de chiffrage de données) possédant un niveau de difficulté suffisant. Le superviseur 57 décode un code de fonction, avec ou sans données d'entrée, provenant du lecteur-enregistreur de cartes 11 et sélectionne une fonction nécessaire à partir des fonctions de base en vue de l'exécuter.

Pour effectuer ces fonctions, la carte à circuit intégré 1 comprend un élément de commande (par exemple une CPU) 59, qui fait fonction de dispositif de commande, une mémoire de données

permanente 61 dont le contenu peut être effacé, une mémoire de programmes 63, et un moyen de contact 65 servant à assurer le contact électrique avec le lecteur-enregistreur de cartes 11, comme représenté sur la figure 9. Parmi ceux-ci, les composants entourés d'une ligne en trait interrompu (la CPU 59, la mémoire de données 61 et la mémoire de programmes 63) sont constitués d'une unique puce IC. On note que la CPU 59 contient la mémoire vive, ou RAM, 60. La mémoire de programmes 63 comprend une mémoire morte, ou ROM à masque, et emmagasine un programme de commande de la CPU 59 possédant des sous-programmes destinés à effectuer les fonctions de base ci-dessus mentionnées. La mémoire de données 61 comprend une mémoire morte programmable électriquement effaçable, ou EEPROM, servant à emmagasiner diverses données.

La mémoire de données 61 est divisée dans plusieurs zones 01, 02, ..., FF, comme représenté sur la figure 10, et chaque zone est divisée en un ou plusieurs blocs. Chaque bloc possède un nombre prédéterminé de bytes, et le traitement s'exécute en unités de blocs. Chaque bloc est constitué d'une donnée d'attribut (un seul byte) et d'une donnée d'emmagasinage. Comme représenté sur la figure 10, les numéros des zones [00] à [FF] sont assignés aux zones respectives (la zone 00 n'étant pas représentée). Dans la zone [00], qui est représentée sur la figure 11, le nombre de bytes 64, l'adresse de début 66 de chaque zone, et l'adresse finale 68 de chaque zone sont emmagasinés en correspondance avec le numéro de zone 62 (ainsi que cela sera explicité ci-après en relation avec une table d'index). Par exemple, l'adresse de début de la zone [01] est aaa et son adresse finale est bbb. Chaque bloc de la zone [01] possède six bytes (la donnée d'attribut possède un byte et la donnée d'emmagasinage possède cinq bytes). Au début de chaque zone, une région servant à emmagasiner l'adresse du byte final du bloc final de la donnée écrite (ci-après appelée une donnée de pointeur) est prévue. Par exemple, à deux bytes à partir de l'adresse aaa de la zone [01] sur la figure 10, se trouve une adresse qui est celle à laquelle le byte final du bloc final de la donnée écrite est

écrit. Sur la figure 10, les parties hachurées désignent des parties dans lesquelles est emmagasinée la donnée d'attribut relative au bloc correspondant. La donnée d'attribut comporte un identificateur qui indique si la donnée d'emmagasinage est ou non valable et un identificateur indiquant si un bloc contient ou non une donnée finale lorsqu'une série de chaînes de données d'emmagasinage sont emmagasinées sur l'étendue de plusieurs blocs. La figure 12 montre le format de la donnée d'attribut. Comme représenté sur la figure 12, le bit "6" est un identificateur indiquant si la donnée d'emmagasinage dans un bloc est ou non valable. Si une donnée "1" est fixée dans le bit "6", ceci indique que la donnée d'emmagasinage n'est pas valable, et si une donnée "0" y est placée, ceci indique que la donnée est valable. Le bit "7" est un identificateur indiquant si un bloc comporte ou non le byte final d'une série de données. Si la donnée "1" est placée dans ce bit, ceci indique que le bloc ne contient pas le byte final et, si la donnée "0" y est placée, ceci indique que le bloc comporte le byte final. On note que les bits "0" à "5" sont des bits fictifs.

On va maintenant décrire l'opération d'écriture de données dans la mémoire de données 61 à l'aide de la disposition ci-dessus présentée en relation avec les organigrammes des figures 13A et 13B. Lors de l'écriture d'une donnée dans la mémoire de données 61, une donnée d'instruction d'écriture présentant le format indiqué sur la figure 14 est appliquée en entrée. La donnée d'instruction d'écriture est constituée d'un code de fonction d'écriture 70, d'un numéro de zone 72 et de données d'emmagasinage. Les données d'emmagasinage sont constituées d'une chaîne de données 76 à emmagasiner et de la donnée du nombre de bytes 74 constituant la chaîne de données. Dans l'état normal, la CPU 59 attend une donnée d'instruction de la part du lecteur-enregistreur de cartes 11. Lorsque la CPU 59 reçoit une donnée d'instruction de la part du lecteur-enregistreur 11, elle vérifie au cours de l'étape 71 si le code de fonction inclus dans la donnée d'instruction d'entrée est un code de fonction d'écriture. Si la réponse est OUI à l'étape 71, la CPU 59 recherche le numéro

de zone ajouté à la donnée d'instruction dans la zone [00] au cours de l'étape 73. Si la réponse est NON à l'étape 73, la CPU 59 délivre une donnée de réponse indiquant que l'aire correspondante n'a pas été trouvée au cours de l'étape 77, puis attend la donnée d'instruction suivante. Si la réponse est OUI à l'étape 73, c'est-à-dire si le numéro de zone inclus dans la donnée d'instruction est trouvé dans la table d'index, la CPU 59 se reporte au nombre de bytes écrit dans la table d'index. Pour l'écriture de données d'emmagasinage dans une aire désignée, la CPU 59 se rapporte d'abord à la donnée de pointeur se trouvant au début de l'aire désignée afin de certifier l'adresse de début d'écriture au cours de l'étape 75. Ainsi, les données s'écrivent à partir de l'adresse obtenue par incrémentations unitaires de la donnée de pointeur. La CPU 59 vérifie à l'étape 79 si toutes les données d'emmagasinage d'entrée peuvent être écrites dans la zone totale. Cette opération s'effectue en relation avec les adresses de début et de fin de la table d'index. Si la réponse est NON à l'étage 79, la CPU 59 délivre une donnée de réponse indiquant que la donnée du nombre de bytes se trouvant dans la donnée d'instruction est erronée au cours de l'étape 85, puis attend la nouvelle donnée d'instruction. Toutefois, si la réponse est OUI à l'étape 79, la CPU 59 compare la donnée du nombre de bytes avec le nombre de bytes constituant la chaîne de données au cours de l'étape 81. Normalement, la donnée du nombre de bytes incluse dans la donnée d'instruction ne peut pas être transmise par une seule opération de transmission (entre un système principal et la carte à circuit intégré). Par exemple, la donnée du nombre de bytes indique onze bytes et le nombre de bytes des données d'emmagasinage envoyées par une seule donnée d'instruction est cinq bytes. Si la réponse est OUI à l'étape 81, c'est-à-dire si la donnée du nombre de bytes est plus petite que le nombre de bytes de la chaîne de données, la CPU 59 délivre une donnée de réponse indiquant que la donnée du nombre de bytes est erronée au cours de l'étape 85, puis attend la donnée d'instruction suivante. Sinon, la CPU 59 soustrait le nombre de bytes contenus dans la chaîne de données de la donnée de nombre de bytes afin d'obtenir une quantité restante et elle

emmagasine la quantité restante dans une zone spécifique de la mémoire de données 61, au cours de l'étape 83.

A l'étape 87, la CPU 59 positionne alors à "1" le bit le plus significatif (MSB) d'une chaîne de bits constituant la donnée du nombre de bytes par la chaîne de données, ce qui rend temporairement non valable la donnée d'emmagasinage. La CPU 59 divise la donnée d'emmagasinage en unités de traitement au cours de l'étape 89, ajoute une donnée d'attribut à chaque unité de traitement, et l'emmagasine dans une aire de donnée désignée (étapes 91, 93 et 95). La CPU 59 vérifie à l'étape 91 si la chaîne de données à écrire est la donnée divisée finale. Si la réponse est OUI à l'étape 91 et s'il est déterminé à l'étape 97 que la donnée "0" est positionnée dans le bit indiquant la quantité restante, toute la donnée finale du bloc final est alors emmagasinée. De plus, le MSB de la chaîne de bits constituant la donnée du nombre de bytes de la chaîne de données est positionné à "0" d'une manière à rendre valables les données à emmagasiner, tandis que l'adresse finale du bloc comportant le byte final de la chaîne de donnée est emmagasinée comme donnée de pointeur (étapes 99 et 101). La CPU 59 délivre une donnée de réponse indiquant l'achèvement de l'écriture au cours de l'étape 103, puis attend la nouvelle donnée d'instruction. Toutefois, s'il est déterminé à l'étape 97 que le bit indiquant la quantité restante n'est pas "0", la CPU 59 emmagasine la donnée divisée finale de la chaîne de donnée incluse dans la donnée d'instruction dans la RAM interne 60 sans l'écrire dans la zone désignée au cours de l'étape 105. Ensuite, la CPU 59 positionne un drapeau d'acceptation de continuation d'écriture dans la RAM 60, et l'adresse de début du bloc non écrit suivant dans la RAM 60 comme adresse de début d'écriture au cours de l'étape 107. La CPU 59 délivre une donnée de réponse indiquant l'acceptation de continuation d'écriture au cours de l'étape 109, puis attend la donnée d'instruction suivante.

On va maintenant décrire une opération d'écriture en continuation, ou "à suivre", à l'aide des organigrammes des figures 15A et 15B. Lorsqu'une opération d'écriture à suivre est

exécutée, une donnée d'instruction d'écriture à suivre possédant le format présenté sur la figure 16 est appliquée en entrée. La donnée d'instruction d'écriture à suivre est constituée d'un code de fonction d'écriture à suivre et de données d'emmagasinage. Si la donnée d'instruction d'écriture à suivre est reçue au cours de l'étape 111, la CPU 59 se reporte au drapeau d'acceptation de continuation d'écriture afin de vérifier, au cours de l'étape 113, si celui-ci est positionné. Si la réponse est NON à l'étape 113, la CPU 59 délivre une donnée de réponse indiquant une erreur de séquence au cours de l'étape 115 et attend la donnée d'instruction suivante. Toutefois, si la réponse est OUI à l'étape 113, la CPU 59 compare le nombre de bytes de la donnée d'emmagasinage d'entrée avec la quantité restante emmagasinée dans la RAM 60 au cours de l'étape 117. Si le nombre de bytes est plus grand que la quantité restante, la CPU 59 délivre une donnée de réponse indiquant que la donnée du nombre de bytes est erronée au cours de l'étape 19, puis attend la donnée d'instruction suivante. Sinon, la CPU 59 soustrait le nombre de bytes de la quantité restante au cours de l'étape 121 et emmagasine la différence comme nouvelle quantité restante.

La CPU 59 adjoint la donnée divisée finale précédente à l'amorce de la donnée d'emmagasinage d'entrée afin d'obtenir une nouvelle donnée d'emmagasinage au cours de l'étape 123 et elle divise celle-ci en unités de traitement et ajoute une donnée d'attribut à chaque unité de traitement au cours de l'étape 125. La CPU 59 emmagasine ensuite la donnée en fonction de l'adresse de début d'écriture emmagasinée dans la RAM 60 (étapes 127, 129 et 131). Dans ce cas, la CPU 59 vérifie à l'étape 127 si la chaîne de données à écrire est la donnée divisée finale. Si la réponse est OUI à l'étape 127, le programme passe à l'étape 133. Si la donnée est OUI à l'étape 133, c'est-à-dire si le bit indiquant la quantité restante est "0", toutes les données d'emmagasinage sont emmagasinées, le MSB de la donnée du nombre de bytes dans la chaîne de données est positionné à "0", et l'adresse finale du bloc comportant le byte final de la chaîne de données d'emmagasinage est emmagasinée comme donnée de pointeur

(étapes 135 et 137). La CPU 59 repositionne le drapeau d'acceptation de continuation d'écriture au cours de l'étape 139, délivre une donnée de réponse indiquant l'achèvement de l'écriture au cours de l'étape 141, puis attend la donnée d'instruction suivante. Toutefois, si la réponse est NON au cours de l'étape 133, la CPU 159 emmagasine la donnée divisée finale dans la RAM 60 sans l'écrire dans la zone spécifiée au cours de l'étape 143. A l'étape 145, la CPU 59 positionne le drapeau d'acceptation de continuation d'écriture et emmagasine l'adresse de début du bloc non écrit dans la RAM 60 comme adresse de début d'écriture. A l'étape 147, la CPU 59 délivre une donnée de réponse indiquant l'acceptation de la continuation d'écriture, puis attend la donnée d'instruction suivante.

Une chaîne de données qui est trop longue pour être emmagasinée par une seule transmission peut être emmagasinée comme ci-dessus décrit. Lorsque la chaîne de données est emmagasinée en unités de blocs, dans la donnée d'attribut qui lui a été ajoutée, le bit "6" est positionné à "0", et le bit "7" est positionné à "1". Lorsque le bit indiquant la quantité restante est "0", dans un bloc emmagasinant la donnée divisée finale, les deux bits "6" et "7" sont positionnés à "0".

On suppose que la donnée d'instruction représentée sur la figure 17A est appliquée en entrée. La donnée d'instruction présentée sur la figure 17A est une donnée d'instruction d'écriture, et la zone cible est [02]. Le nombre de bytes de la zone [02] est quatre bytes, comme indiqué dans la table d'index de la figure 11. Tout d'abord, il est effectué une extraction de la donnée du nombre de bytes dans la donnée d'instruction d'entrée, puis le programme se reporte à la donnée de pointeur placée au début de la zone [02] afin de vérifier si toutes les données d'emmagasinage peuvent être emmagasinées. Ensuite, le MSB de la donnée du nombre de bytes de la donnée d'instruction est positionné à "1" (figure 17B). Comme le montre la figure 17B, "80" et "0B" sont exprimés en notation hexadécimale. La quantité restante déterminée en fonction du nombre de bytes à partir de la table d'index et de la donnée du nombre de bytes se trouvant

dans la donnée d'instruction (figure 17C). Dans ce cas, puisque la donnée du nombre de bytes est "11" et que le nombre de bytes positionné dans la donnée d'instruction est "5", la quantité restante correspond à "6". Cinq bytes sont divisés par le nombre de bytes de l'unité de traitement (figure 17D) et sont emmagasinés dans la zone [02], comme représenté sur la figure 17E. Puisque la quantité restante n'est pas "0", la donnée divisée finale n'est pas emmagasinée. Ensuite, l'adresse d'écriture suivant est emmagasinée et le drapeau d'acceptation de continuation d'écriture incorporé dans la mémoire de données 61 est positionné.

Dans cet état, si la donnée d'instruction d'écriture à suivre présentée sur la figure 17F est appliquée en entrée, le nombre de bytes contenus dans la donnée d'instruction subit une vérification. Une nouvelle quantité restante est positionnée en fonction de la quantité restante précédente et du nombre de bytes de la donnée d'emmagasinage appliquée en entrée (figure 17G). Dans ce cas, puisque la quantité restante précédente correspond à "6" et que le nombre de bytes de la donnée d'emmagasinage d'entrée correspond à "6", la nouvelle quantité restante correspond à "0". La donnée divisée finale préalablement emmagasinée et la donnée d'emmagasinage appliquée en entrée sont combinées (figure 17H), la donnée d'emmagasinage est divisée par le nombre de bytes de l'unité de traitement (figure 17I), et elle est emmagasinée aux adresses correspondantes de la zone [02] (figure 17J). Dans ce cas, puisque la quantité restante correspond à "0", le MSB de la donnée du nombre de bytes est positionné à "0", et l'adresse finale est emmagasinée comme donnée de pointeur.

On va maintenant décrire l'opération de lecture d'une donnée dans la mémoire de données 61 en relation avec les organigrammes des figures 18A et 18C. Lors de la lecture d'une donnée emmagasinée dans la mémoire de données 61, une donnée d'instruction de lecture possédant le format présenté sur la figure 19 est appliquée en entrée. La donnée d'instruction de lecture est constituée d'un code de fonction de lecture et d'un numéro de zone. Dans l'état normal, la CPU 59 attend une donnée d'instruction de la part du lecteur-enregistreur de cartes 11.

Lorsque la donnée d'instruction du lecteur-enregistreur 11 a été appliquée, la CPU 59 vérifie à l'étape 149 si le code de fonction appartenant à la donnée d'instruction d'entrée est un code de fonction de lecture. Si la réponse est OUI à l'étape 149, la CPU 59 trouve le numéro de zone dans la donnée d'instruction à partir de la zone [00] de la mémoire de données 61 au cours de l'étape 151. Si la réponse est NON à l'étape 151, la CPU 59 délivre une donnée de réponse indiquant que la zone correspondante n'a pas été trouvée au cours de l'étape 153, puis attend la donnée d'instruction suivante. Si la réponse est OUI à l'étape 151, la CPU 59 se reporte au nombre de bytes de traitement correspondant à la zone détectée et emmagasine les adresses de début et de fin de la zone correspondante dans la mémoire de données 61. Lorsque la donnée se trouvant dans la zone est lue, la CPU 59 se reporte à la donnée de pointeur placée au début de la zone pour vérifier au cours de l'étape 155 si tous les bits de la donnée de pointeur valent "1". Lorsque la CPU 59 certifie qu'aucune donnée n'a été emmagasinée dans cette zone, elle délivre une donnée de réponse indiquant une zone non écrite au cours de l'étape 157, puis attend la nouvelle donnée d'instruction. Toutefois, si la réponse est NON à l'étape 155, c'est-à-dire si tous les bits de la donnée de pointeur ne valent pas "1", la CPU 59 trouve le bloc de début dans la zone à partir de la donnée de pointeur au cours de l'étape 159. Puisque le bloc de début emmagasine la donnée du nombre de bytes, la CPU 59 extrait la donnée du nombre de bytes au cours de l'étape 161 pour vérifier si la chaîne de données visée peut être présente dans la zone. Par exemple, si la donnée du nombre de bytes écrite dans le bloc de début est plus grande qu'une capacité d'emmagasinage calculée à partir des adresses de début et de fin de la table d'index, ceci signifie une erreur. En résultat, si une erreur est détectée, la CPU 59 délivre une donnée de réponse indiquant que la donnée du nombre de bytes est erronée au cours de l'étape 163, puis attend la nouvelle donnée d'instruction. Toutefois, si la réponse est OUI à l'étape 161, c'est-à-dire si la donnée du nombre de bytes est valable, la CPU 59 positionne la donnée du nombre de bytes extraite comme valeur initiale d'un compteur du type logiciel emmagasiné dans la RAM 60. A ce

moment, s'il est déterminé au cours de l'étape 165 que le MSB de la donnée du nombre de bytes vaut "1", la CPU 59 certifie que la chaîne de données suivante n'est pas une donnée valable et positionne le drapeau de présence d'une donnée non valable au cours de l'étape 167. La CPU 59 lit alors la chaîne de données après la donnée du nombre de bytes, byte par byte, et l'emmagasine dans la RAM 60. Dans ce cas, le compteur du type logiciel subit une décrémentation unitaire à chaque fois qu'une donnée de un byte est lue. Toutefois, lorsqu'une donnée d'attribut est lue, le compteur du type logiciel n'est pas remis à jour et la donnée d'attribut n'est pas emmagasinée dans la RAM 60.

Les données d'emmagasinage s'emmagasinent dans la RAM 60 jusqu'à ce que le contenu du compteur atteigne "0" (étapes 169, 171, 173, 175 et 179). Toutefois, puisque la RAM 60 a une capacité limitée, si le nombre de bytes des données atteint la capacité avant que le contenu du compteur ait atteint "0", la CPU 59 emmagasine le contenu en vigueur du compteur et l'adresse finale de la donnée en cours de lecture et, en même temps positionne un drapeau d'acceptation de continuation de lecture dans la RAM 60 au cours de l'étape 183. La CPU 59 délivre une donnée de réponse indiquant l'acceptation de la continuation de la lecture en même temps que la chaîne de données emmagasinée dans la RAM au cours de l'étape 185, puis attend la donnée d'instruction suivante. Si le contenu du compteur logiciel a atteint "0", la CPU 59 vérifie au cours de l'étape 187 si le drapeau de présence d'une donnée non valable est positionné. Si la réponse est NON à l'étape 187, la CPU 59 délivre une donnée de réponse en même temps que la chaîne de données emmagasinée dans la RAM 60 au cours de l'étape 189, puis attend la donnée d'instruction suivante. Si la réponse est OUI à l'étape 187, la CPU 59 délivre une donnée de réponse indiquant une donnée non valable avec la chaîne de données emmagasinée dans la RAM 60 au cours de l'étape 191, puis attend la donnée d'instruction suivante.

On va maintenant décrire une opération de lecture à suivre en relation avec l'organigramme des figures 20A et 20B. Lors de l'exécution d'une opération de lecture à suivre, une

donnée d'instruction de lecture à suivre possédant le format présenté sur la figure 20C est délivrée. La donnée d'instruction de lecture à suivre est constituée d'un code de fonction de lecture à suivre et d'un numéro de zone. Si la donnée d'instruction de lecture à suivre est reçue au cours de l'étape 193, la CPU 59 se reporte d'abord à un drapeau d'acceptation de continuation de lecture pour vérifier au cours de l'étape 195 si celui-ci est positionné. Si la réponse est NON à l'étape 195, la CPU 59 délivre une donnée de réponse indiquant une erreur de séquence au cours de l'étape 197, puis attend la donnée d'instruction suivante. Si la réponse est OUI à l'étape 195, la CPU 59 lit les données, byte par byte, suivant l'adresse précédemment emmagasinée tout en décrémentant d'une unité le compteur de type logiciel et elle emmagasine les données dans la RAM 60 (opérations 199, 201 et 203). Toutefois, lors de la lecture de la donnée d'attribut, le compteur du type logiciel n'est pas remis à jour et la donnée d'attribut n'est pas emmagasinée dans la RAM 60. Si le contenu du compteur atteint "0" à l'étape 207, la CPU 59 repositionne le drapeau d'acceptation de continuation de lecture au cours de l'étape 211 et vérifie, au cours de l'étape 219, si le drapeau de présence d'une donnée non valable est positionné. Si la réponse est NON à l'étape 219, la CPU 59 délivre une donnée de réponse indiquant l'achèvement de la lecture, en même temps que la chaîne de données maintenue dans la RAM 60 au cours de l'étape 221, puis attend la donnée d'instruction suivante. Si la réponse est OUI à l'étape 219, la CPU 59 délivre une donnée de réponse indiquant une donnée non valable, en même temps que la chaîne de données maintenue dans la RAM 60 au cours de l'étape 217, puis attend la donnée d'instruction suivante. Si la capacité de la RAM 60 est atteinte avant que le contenu du compteur n'ait atteint "0" au cours de l'étape 205, la CPU 59 emmagasine le contenu du compteur et l'adresse finale de l'opération de lecture de donnée en cours et, dans le même temps, positionne le drapeau d'acceptation de continuation de lecture au cours de l'étape 213. La CPU 59 délivre ensuite une donnée de réponse indiquant une acceptation de continuation de lecture, en même temps que la chaîne de données dans

la RAM 60 au cours de l'étape 215, puis attend la donnée d'instruction suivante.

Plus spécialement, on suppose que l'opération de lecture s'effectue en réponse à la donnée d'instruction de lecture représentée sur la figure 21A, relativement à une zone telle que présentée sur la figure 17J (par exemple la zone [02]). On note que la capacité de la RAM 60 est huit bytes. Dans ce cas, une chaîne de données lue par cette donnée d'instruction est une chaîne de données telle que représentée sur la figure 21B, et la donnée du nombre de bytes indique onze bytes. Ainsi, lorsque la chaîne de données lue est emmagasinée dans la RAM 60, le contenu du compteur logiciel correspond à "5". Puisque le contenu du compteur n'est pas "0", une donnée de réponse indiquant une acceptation de continuation de lecture est délivrée en même temps que la chaîne de données (figure 21C). Lorsque, ensuite, la donnée d'instruction de lecture à suivre représentée sur la figure 21D est reçue, la chaîne de données représentée sur la figure 21E est emmagasinée dans la RAM 60. A ce moment, puisque le contenu du compteur correspond à "0", une donnée de réponse indiquant l'achèvement de la lecture est délivrée en même temps que la chaîne de données (figure 21F).

On va décrire une opération d'effacement de donnée dans la mémoire de données 61 en relation avec l'organigramme de la figure 22. Lorsqu'une donnée emmagasinée dans la mémoire de données 61 est effacée, une donnée d'instruction d'effacement ayant le format présenté sur la figure 23 est appliquée en entrée. La donnée d'instruction d'effacement est constituée d'un code de fonction d'effacement et d'un numéro de zone. Dans l'état normal, la CPU 59 attend une donnée d'instruction de la part du lecteur-enregistreur de cartes 11. Lorsqu'une donnée d'instruction est délivrée par le lecteur-enregistreur de cartes 11 à cet instant, la CPU 59 vérifie que cours de l'étape 223 si le code de fonction se trouvant dans la donnée d'instruction est un code de fonction d'effacement. Si la réponse est OUI à l'étape 223, la CPU 59 recherche le numéro de zone appartenant à la donnée d'instruction dans la zone [00] de la mémoire de données 61 au cours

de l'étape 225. Si la réponse est NON à l'étape 225, la CPU 59 délivre une donnée de réponse indiquant que la zone correspondante n'a pas été trouvée au cours de l'étape 227, et elle attend la donnée d'instruction suivante. Si la réponse est OUI au cours de l'étape 225, la CPU 59 se reporte à l'adresse de début de la zone correspondante afin de vérifier la donnée de pointeur qu'elle contient au cours de l'étape 229. En résultat, si tous les bits de la donnée de pointeur valent "1", la CPU 59 détermine qu'aucune donnée n'est emmagasinée dans cette zone, délivre une donnée de réponse indiquant une zone non écrite au cours de l'étape 231, puis attend la donnée d'instruction suivante. Toutefois, si tous les bits de la donnée de pointeur ne sont pas "1", la CPU 59 positionne tous les bits de la donnée de pointeur à "1", délivre une donnée de réponse indiquant l'achèvement de l'opération d'effacement au cours de l'étape 237, puis attend la donnée d'instruction suivante.

Dans la carte à circuit intégré ci-dessus décrite, une zone servant à emmagasiner l'adresse (donnée de pointeur) du byte final du bloc final de la donnée écrite pendant la remise à jour est prévue au début de chaque zone de donnée dans la mémoire de données, et l'accès suivant aux données est effectué en fonction de la donnée de pointeur emmagasinée dans cette zone. Par exemple, lorsque la donnée est écrite dans une zone de donnée cible, l'adresse finale d'une chaîne de données écrite ne doit pas nécessairement être recherchée à partir de la zone cible au contraire de la carte à circuit intégré de fabrication classique, ce qui améliore grandement la vitesse d'accès aux données. De plus, la donnée permettant de certifier une donnée non écrite ne doit pas nécessairement être maintenue, au contraire de la carte à circuit intégré de fabrication classique. On peut effectuer de manière fiable la gestion des données même s'il est adopté un procédé différent d'un procédé de commande de données classique, ce qui autorise une utilisation efficace comme système de carte à circuit intégré.

Dans le mode de réalisation ci-dessus présenté, l'élément de commande, la mémoire de données et la mémoire de

programmes sont constitués par une unique puce IC. Toutefois, ils ne doivent pas nécessairement être constitués par une unique puce IC, mais peuvent être constitués de puces IC distinctes.

Dans le mode de réalisation ci-dessus présenté, la carte à circuit intégré a été illustrée sous forme d'un dispositif électronique portatif. Toutefois, la forme du dispositif électronique portatif ne se limite pas à la forme carte, mais peut avoir la forme d'un bloc ou la forme d'un crayon. Il est possible de modifier la conception matérielle du dispositif électronique portatif en restant dans les limites et dans l'esprit de l'invention.

Bien entendu, l'homme de l'art sera en mesure d'imaginer, à partir du dispositif dont la description vient d'être donnée à titre simplement illustration et nullement limitatif, diverses variantes et modifications ne sortant pas du cadre de l'invention.

REVENDICATIONS

1. Dispositif électronique portatif, caractérisé en ce qu'il comporte :

un moyen de mémoire (4a, 5) servant à emmagasiner des données ; et

une unité centrale de traitement (CPU ; 4) servant à commander sélectivement l'accès pour lecture de données ou l'accès pour écriture de données dans ledit moyen de mémoire en réponse à une instruction délivrée par un système principal connecté audit dispositif électronique portatif, ladite CPU emmagasinant une donnée, qui peut fournir l'adresse d'emmagasinage de la zone non écrite suivante, dans une zone prédéterminée dudit moyen de mémoire (4a) lorsque ladite CPU écrit des données dans ledit moyen de mémoire.

2. Dispositif selon la revendication 1, caractérisé en ce que l'adresse d'emmagasinage est une adresse de début.

3. Dispositif selon la revendication 1, caractérisé en ce que ledit moyen de mémoire (4a, 5) comprend une mémoire permanente (5) et une mémoire non permanente (4a), de sorte que les données sont écrites dans ladite mémoire permanente et l'adresse d'emmagasinage est emmagasinée dans ladite mémoire non permanente.

4. Dispositif selon la revendication 1, caractérisé en ce que ledit moyen de mémoire (61) est divisé en plusieurs zones, et en ce que chacune desdites zones possède un domaine de donnée de position servant à emmagasiner une donnée indiquant la position d'une chaîne de données finale écrite dans la zone correspondante, si bien que, lorsque ladite CPU (59) accède à la zone, la CPU se reporte au domaine de donnée de position de la zone pour certifier un emplacement auquel il doit être accédé dans ledit moyen de mémoire (61).

5. Dispositif selon la revendication 4, caractérisé en ce que chacune desdites zones est en outre divisée en plusieurs blocs, chaque bloc d'une zone unique a une longueur de données prédéterminée, et ladite CPU (59) effectue un traitement de données en utilisant lesdits blocs comme unités de traitement.

6. Dispositif selon la revendication 5, caractérisé en ce que chacun desdits blocs est constitué d'une donnée d'attribut et d'une donnée d'emmagasinage, et ce que la donnée d'emmagasinage est constituée d'une chaîne de données à emmagasiner et d'une donnée indiquant la longueur de la chaîne de données.

7. Dispositif selon la revendication 4, caractérisé en ce que la position de la chaîne de données finale correspond à une position d'adresse d'un bloc final de la chaîne de données finale.

8. Dispositif selon la revendication 6, caractérisé en ce que la donnée d'attribut possède un identificateur indiquant si une donnée d'emmagasinage correspondante est ou non valable et un identificateur indiquant si un bloc comporte ou non une donnée finale, lorsque la chaîne de donnée à emmagasiner doit être emmagasinée sur plusieurs blocs.

9. Dispositif selon la revendication 4, caractérisé en ce que l'une desdites zones possède une table d'index qui emmagazine la longueur de la donnée d'emmagasinage dans chaque bloc des zones restantes et les adresses de début et de fin des zones restantes en correspondance avec les numéros des zones.

10. Dispositif selon la revendication 9, caractérisé en ce que la CPU reçoit une donnée d'instruction délivrée par ledit système principal connecté audit dispositif électronique portatif et consistant en un code de fonction d'écriture, un numéro de zone, une donnée du nombre de bytes et une chaîne de données, se reporte à la table d'index en relation avec le numéro de zone écrit dans la donnée d'instruction et employé comme paramètre permettant d'obtenir la longueur de donnée pour l'unité de traitement, et divise la chaîne de données en unités de traitement pour l'emmagasiner.

11. Dispositif selon la revendication 1, caractérisé en ce que ladite unité centrale de traitement emmagasine une donnée qui peut fournir l'adresse d'emmagasinage de la zone écrite dans une zone prédéterminée du moyen de mémoire dans le seul cas où il est déterminé que la donnée est correctement emmagasinée dans ladite mémoire de données.

FIG. 1

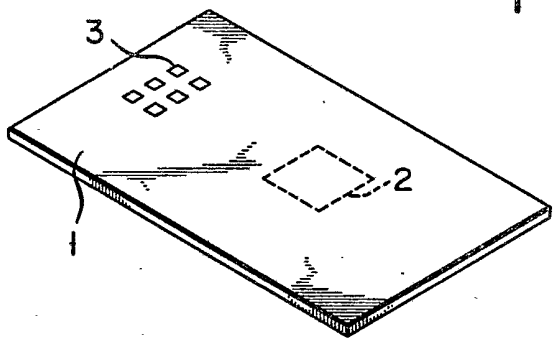


FIG. 2

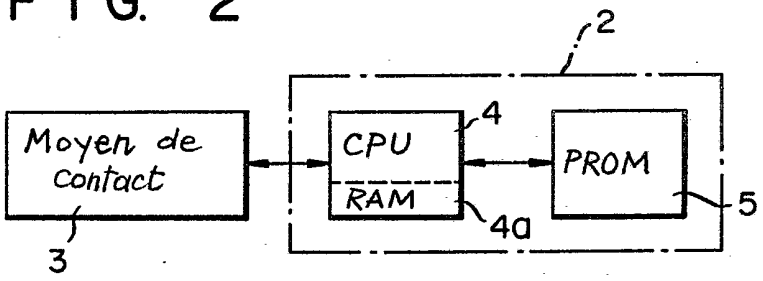


FIG. 3

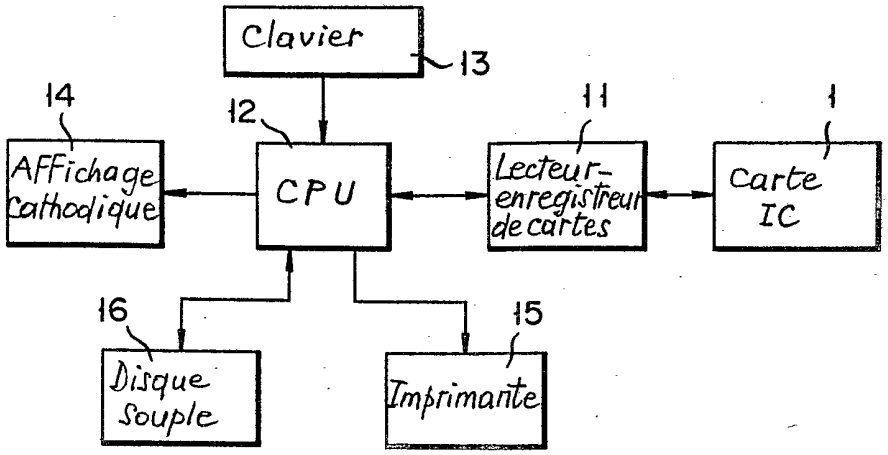


FIG. 4

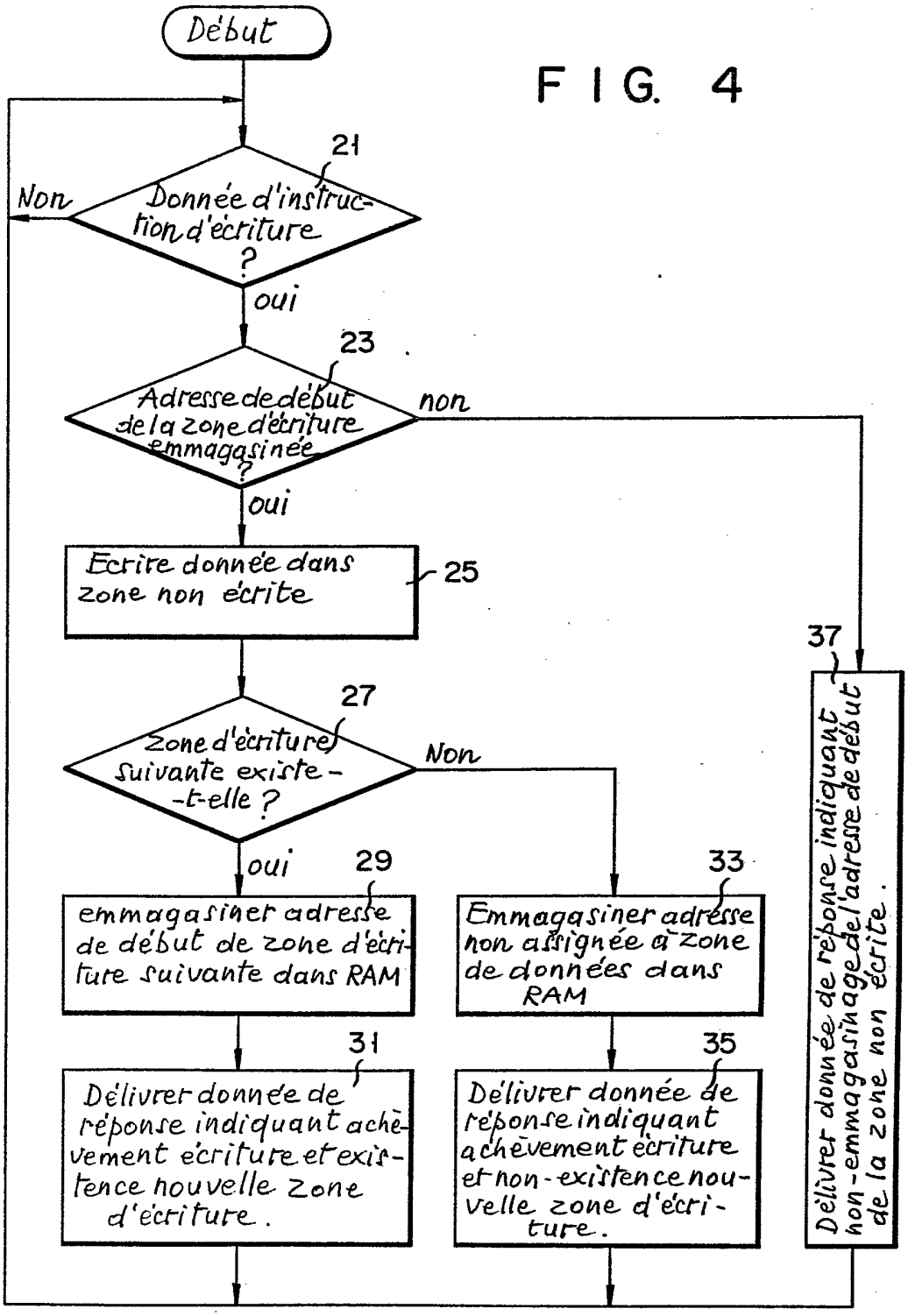

Début
21
Donnée d'instruction d'écriture ?
Non
oui
23
Adresse de début de la zone d'écriture emmagasinée ?
non
oui
Ecrire donnée dans zone non écrite
25
27
Zone d'écriture suivante existe-t-elle ?
Non
oui
29
emmagasiner adresse de début de zone d'écriture suivante dans RAM
31
Délivrer donnée de réponse indiquant achèvement écriture et existence nouvelle zone d'écriture.
33
Emmagasiner adresse non assignée à zone de données dans RAM
35
Délivrer donnée de réponse indiquant achèvement écriture et non-existence nouvelle zone d'écriture.
37
Délivrer donnée de réponse indiquant non-emmagasinage de l'adresse de début de la zone non écrite.

FIG. 5

Code de fonction d'instruction d'écriture	Donnée d'emmagasinage

FIG. 6

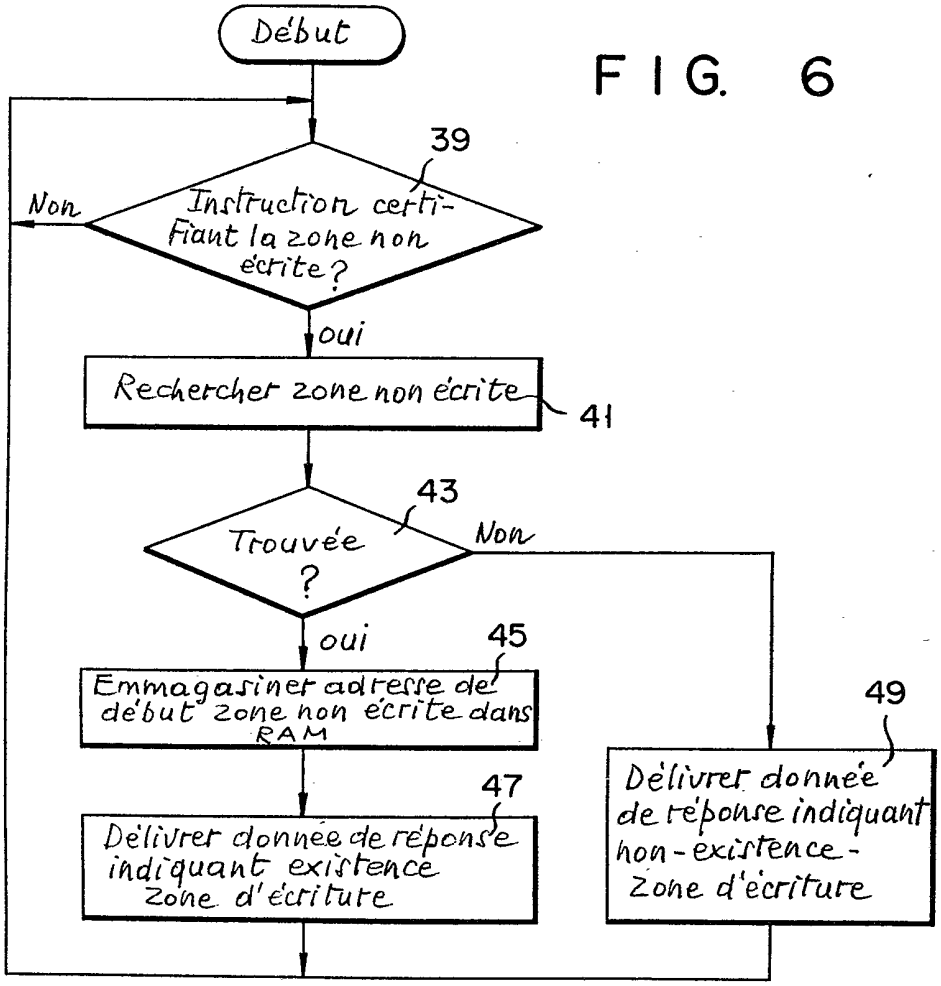


FIG. 7

Certifiant la zone non écrite

FIG. 8

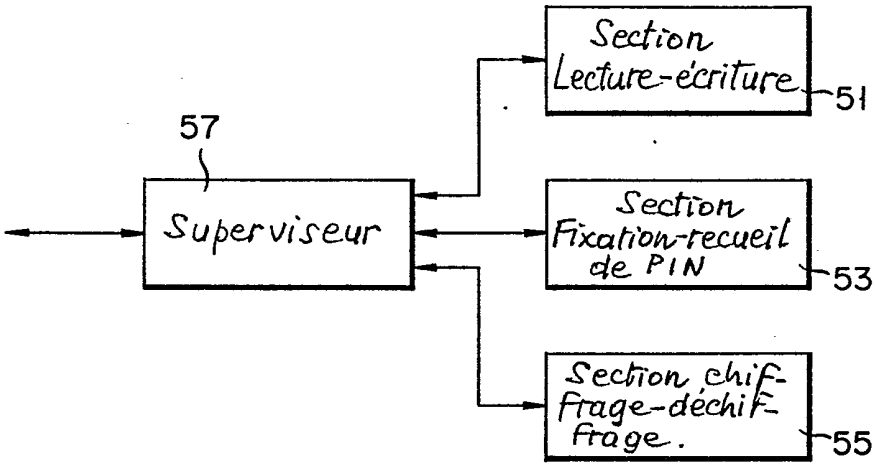

Section Lecture-écriture
51
57
Superviseur
Section Fixation-recueil de PIN
53
Section chiffrage-déchiffrage.
55

FIG. 9

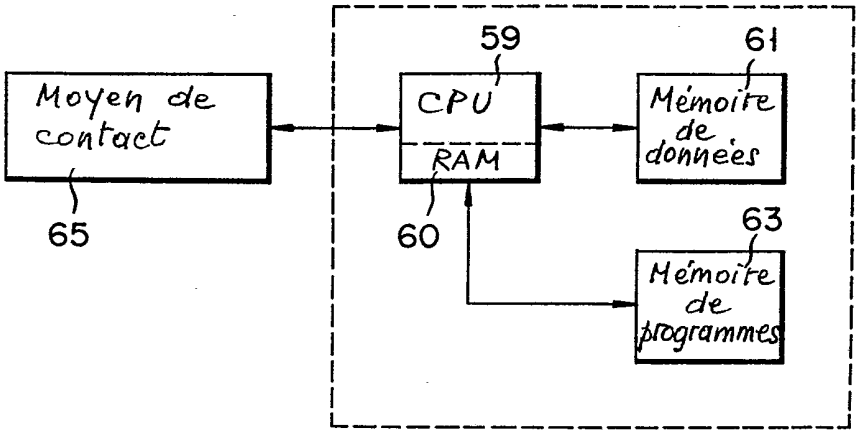

59
61
Moyen de contact
CPU
RAM
Mémoire de données
65
60
63
Mémoire de programmes

FIG. 10

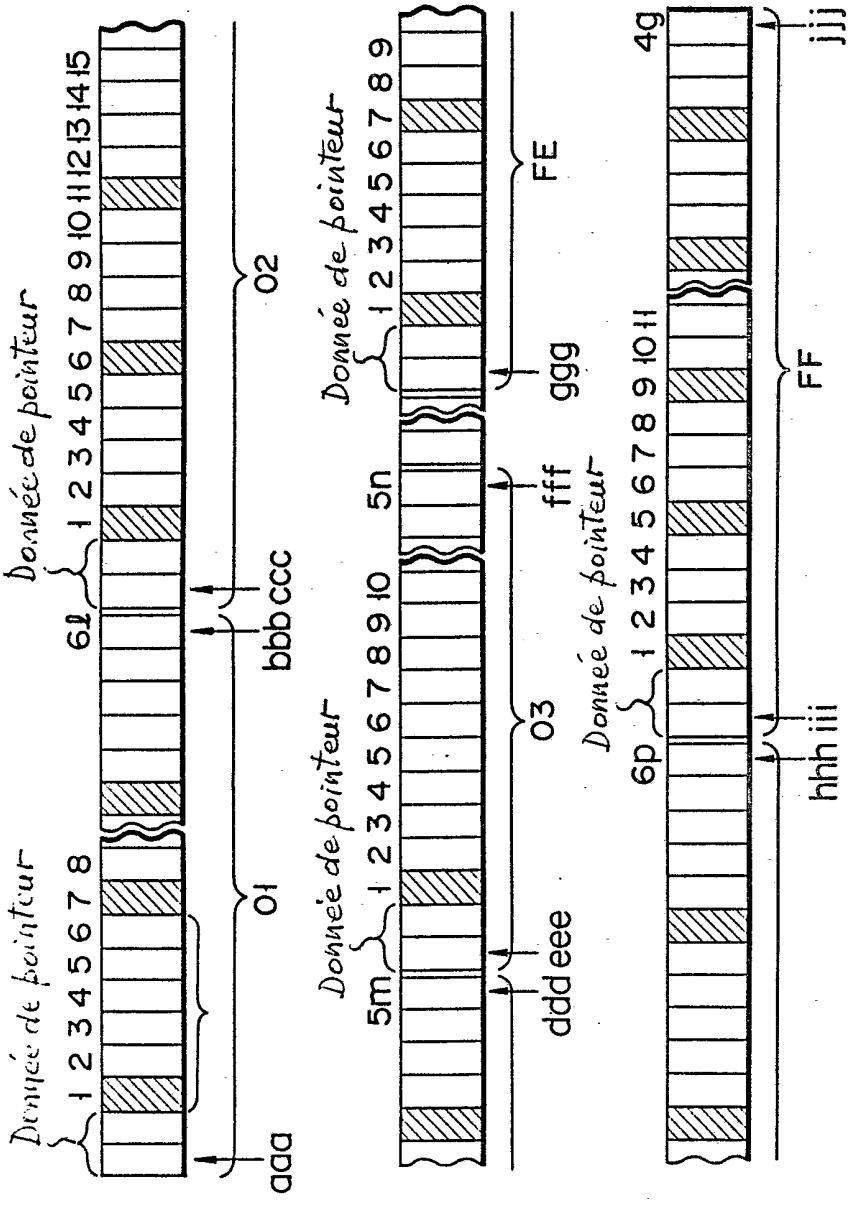

Donnée de pointeur
1 2 3 4 5 6 7 8
aaa
01
Donnée de pointeur
6l
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
bbb ccc
02
Donnée de pointeur
5m
1 2 3 4 5 6 7 8 9 10
ddd eee
03
Donnée de pointeur
5n
1 2 3 4 5 6 7 8 9
fff ggg
FE
Donnée de pointeur
6p
1 2 3 4 5 6 7 8 9 10 11
hhh iii
4g
FF
jjj

FIG. 11

Zone 00

62 Numéro de Zone	64 Nombre de bytes	66 Adresse de début	68 Adresse Finale
01	5	aaa	bbb
02	4	ccc	ddd
03	4	eee	fff
FE	5	ggg	hhh
FF	3	iii	jjj

FIG. 12

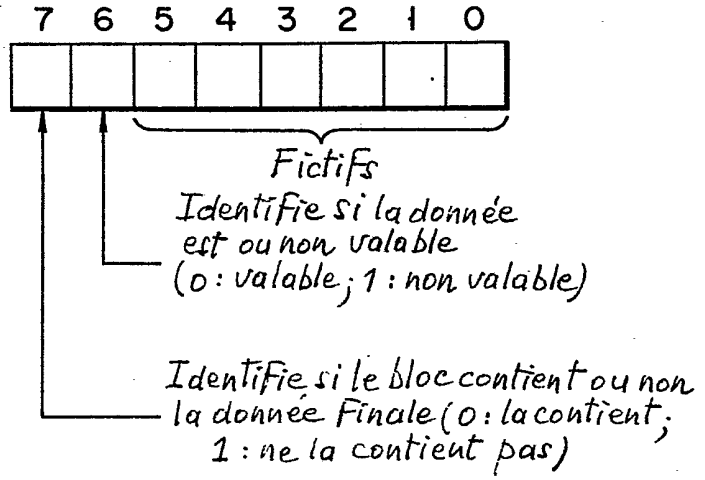


FIG. 13A

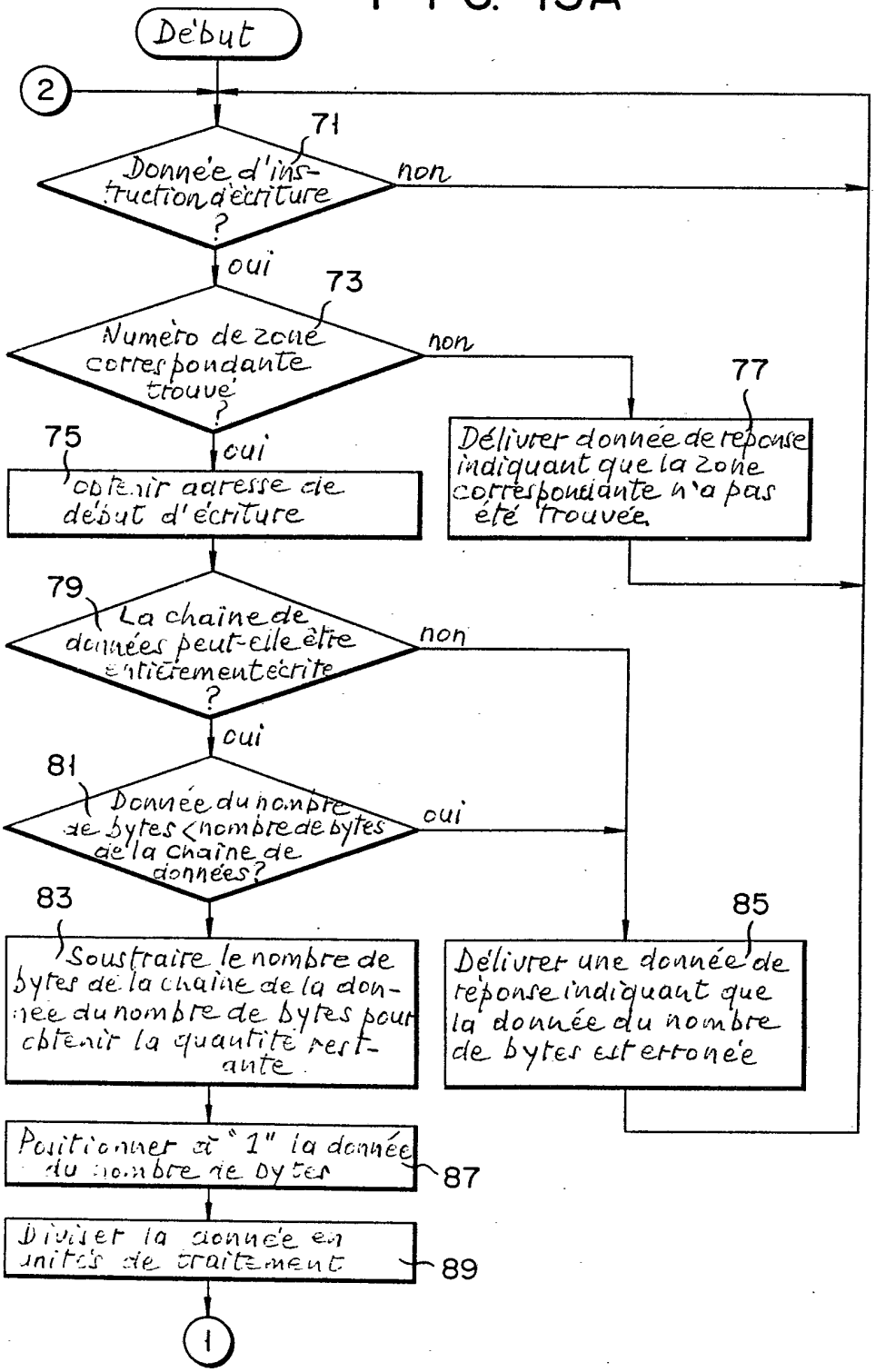

Début
2
71
Donnée d'instruction d'écriture ?
non
oui
73
Numéro de zone correspondante trouvé ?
non
77
Délivrer donnée de réponse indiquant que la zone correspondante n'a pas été trouvée.
oui
75
obtenir adresse de début d'écriture
79
La chaîne de données peut-elle être entièrement écrite ?
non
oui
81
Donnée du nombre de bytes < nombre de bytes de la chaîne de données?
oui
83
Soustraire le nombre de bytes de la chaîne de la donnée du nombre de bytes pour obtenir la quantité restante
85
Délivrer une donnée de réponse indiquant que la donnée du nombre de bytes est erronée
Positionner à "1" la donnée du nombre de bytes
87
Diviser la donnée en unités de traitement
89
1

FIG. 13B

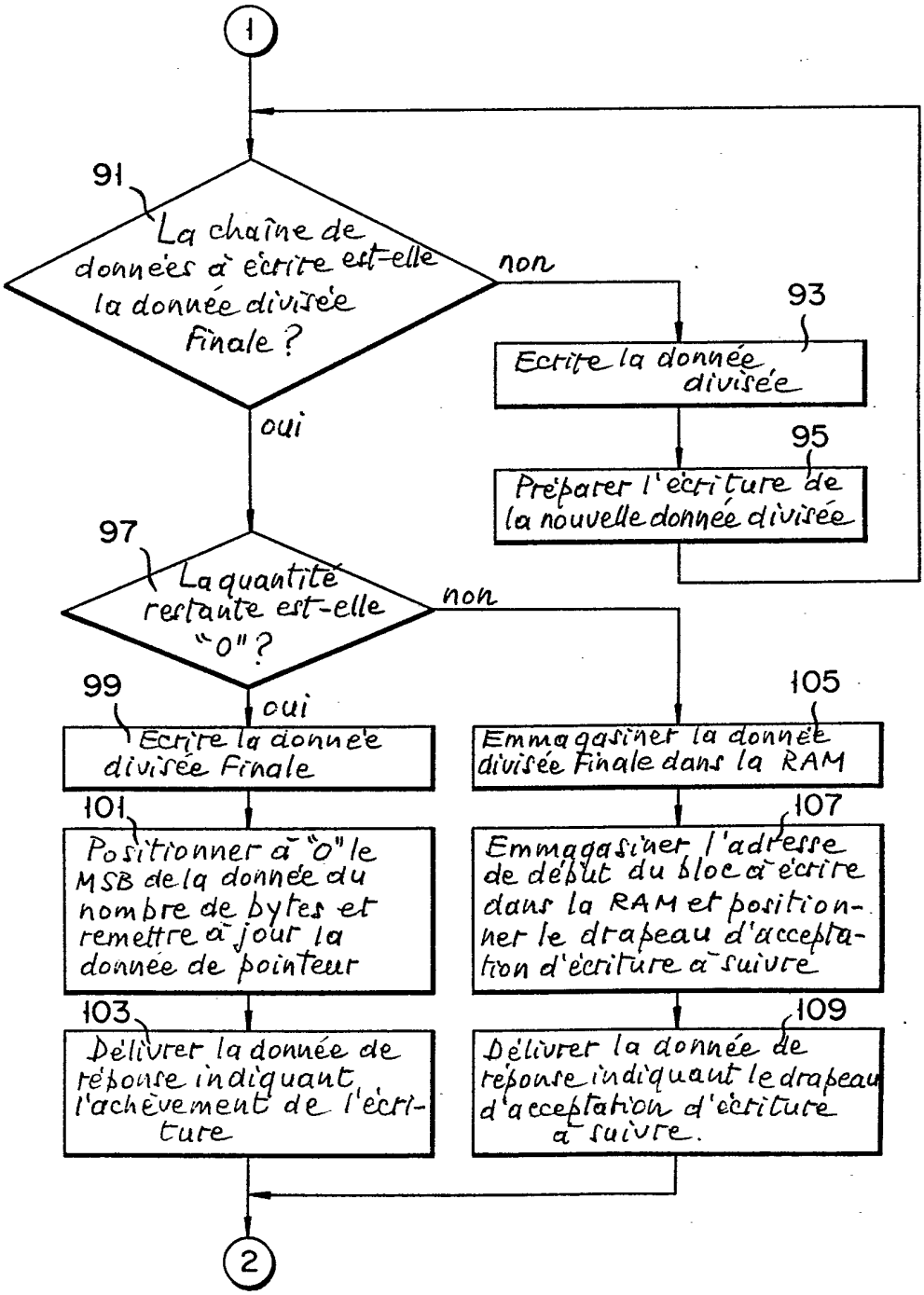

1
91
La chaîne de données à écrire est-elle la donnée divisée Finale ?
non
93
Ecrire la donnée divisée
95
Préparer l'écriture de la nouvelle donnée divisée
oui
97
La quantité restante est-elle "0" ?
non
99
Ecrire la donnée divisée Finale
oui
105
Emmagasiner la donnée divisée Finale dans la RAM
101
Positionner à "0" le MSB de la donnée du nombre de bytes et remettre à jour la donnée de pointeur
107
Emmagasiner l'adresse de début du bloc à écrire dans la RAM et positionner le drapeau d'acceptation d'écriture à suivre
103
Délivrer la donnée de réponse indiquant l'achèvement de l'écriture
109
Délivrer la donnée de réponse indiquant le drapeau d'acceptation d'écriture à suivre.
2

FIG. 14

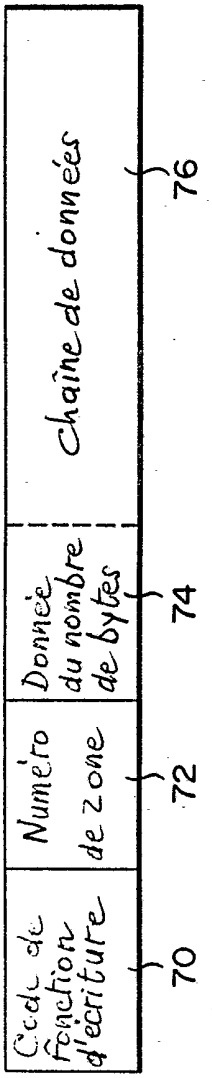

Donnée d'emmagasinage
Code de fonction d'écriture
Numéro de zone
Donnée du nombre de bytes
Chaîne de données
70
72
74
76

FIG. 16

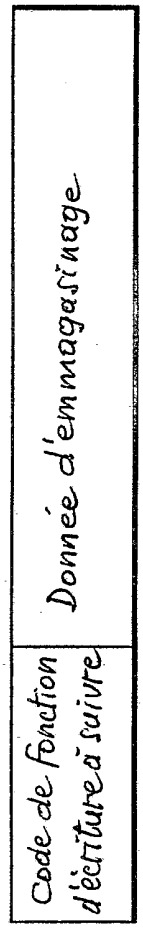

Code de fonction d'écriture à suivre
Donnée d'emmagasinage

FIG. 15A

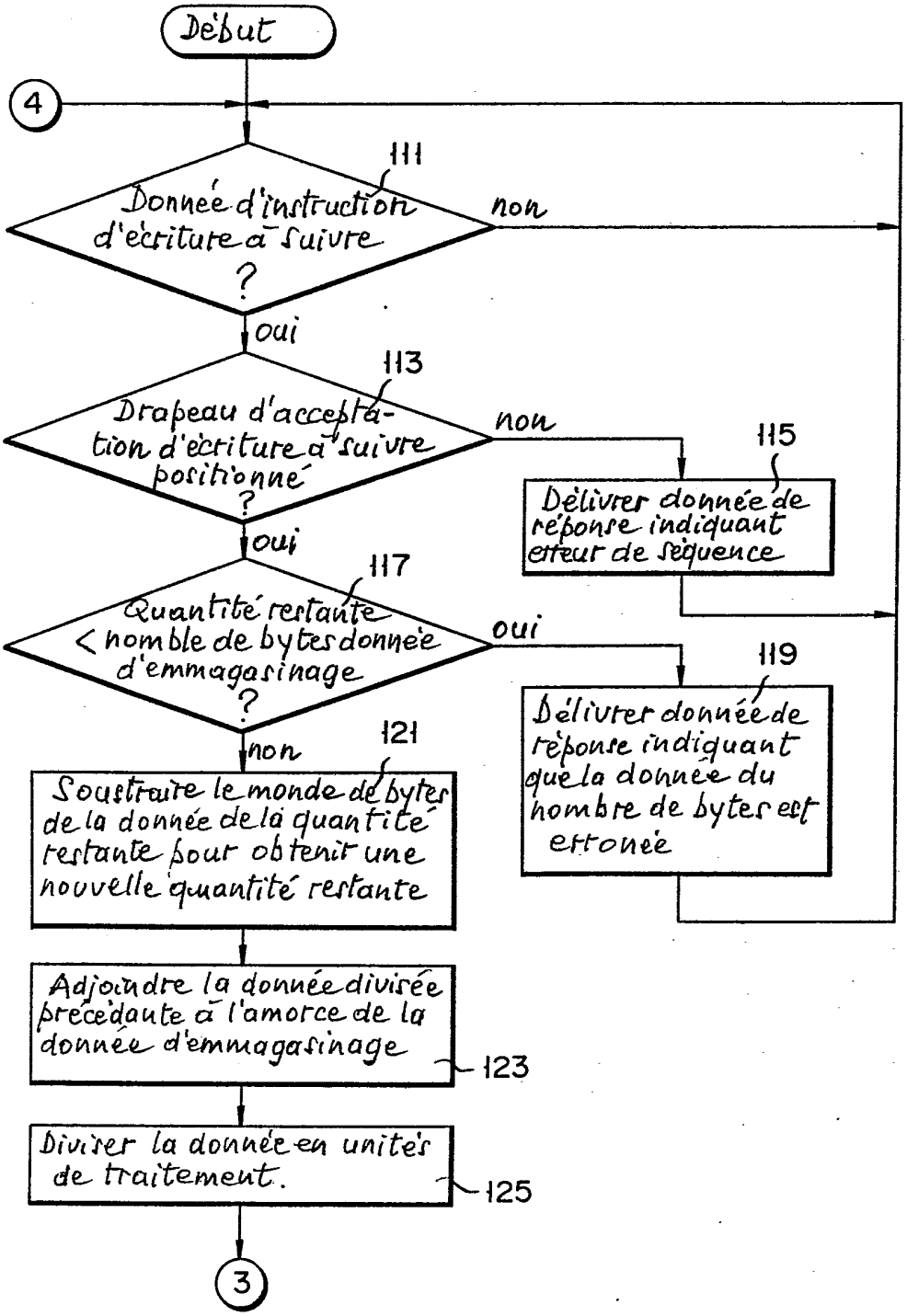

Début
4
111
Donnée d'instruction d'écriture à suivre ?
non
oui
113
Drapeau d'acceptation d'écriture à suivre positionné ?
non
115
Délivrer donnée de réponse indiquant erreur de séquence
oui
117
Quantité restante < nombre de bytes donnée d'emmagasinage ?
oui
119
Délivrer donnée de réponse indiquant que la donnée du nombre de bytes est erronée
non
121
Soustraire le monde de byter de la donnée de la quantité restante pour obtenir une nouvelle quantité restante
Adjoindre la donnée divisée précédante à l'amorce de la donnée d'emmagasinage
123
Diviser la donnée en unités de traitement.
125
3

FIG. 15B

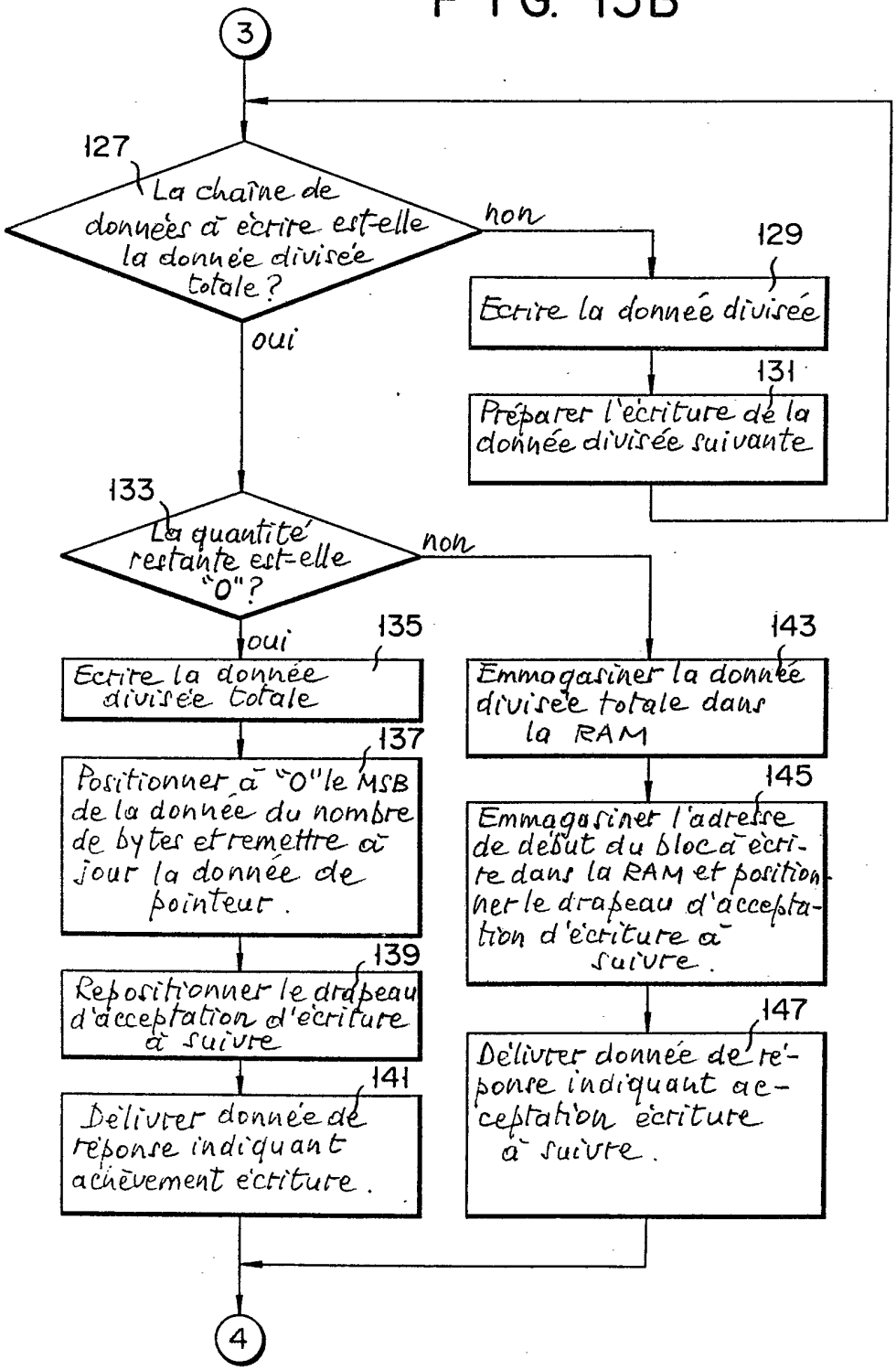

3
127
La chaîne de données à écrire est-elle la donnée divisée totale ?
non
oui
129
Ecrire la donnée divisée
131
Préparer l'écriture de la donnée divisée suivante
133
La quantité restante est-elle "0" ?
non
oui
135
Ecrire la donnée divisée totale
137
Positionner à "0" le MSB de la donnée du nombre de bytes et remettre à jour la donnée de pointeur.
139
Repositionner le drapeau d'acceptation d'écriture à suivre
141
Délivrer donnée de réponse indiquant achèvement écriture.
143
Emmagasiner la donnée divisée totale dans la RAM
145
Emmagasiner l'adresse de début du bloc à écrire dans la RAM et positionner le drapeau d'acceptation d'écriture à suivre.
147
Délivrer donnée de réponse indiquant acceptation écriture à suivre.
4

FIG. 17A

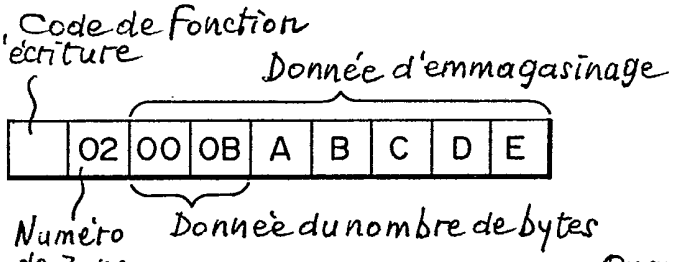


FIG. 17B



FIG. 17C



FIG. 17D



FIG. 17E

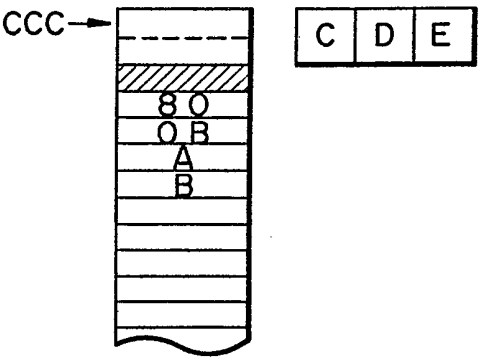


FIG. 17F

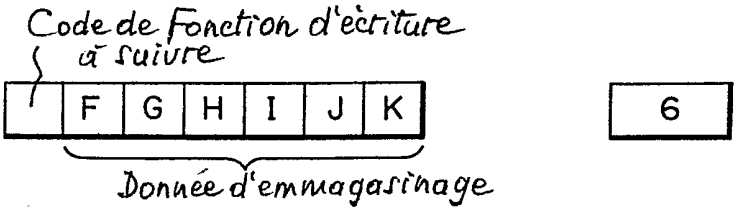


FIG. 17G



FIG. 17H

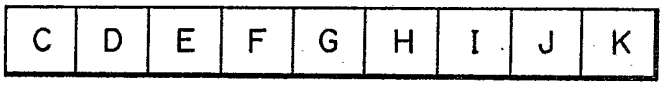


FIG. 17I



FIG. 17J

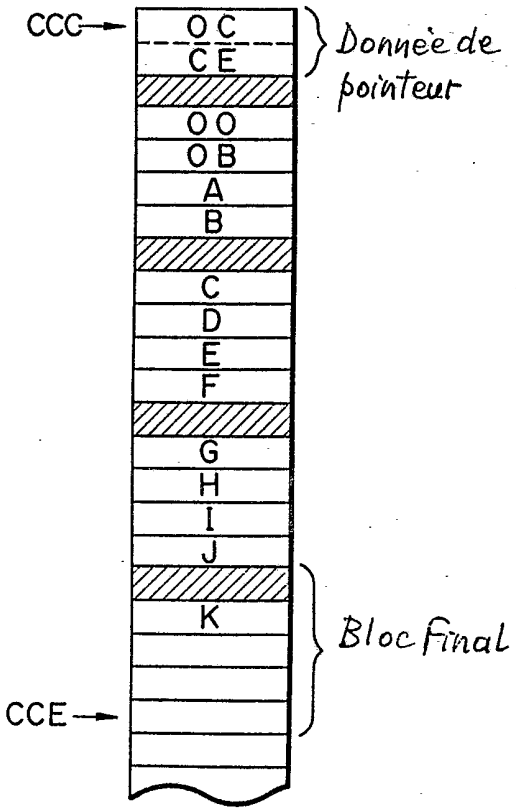


FIG. 18A

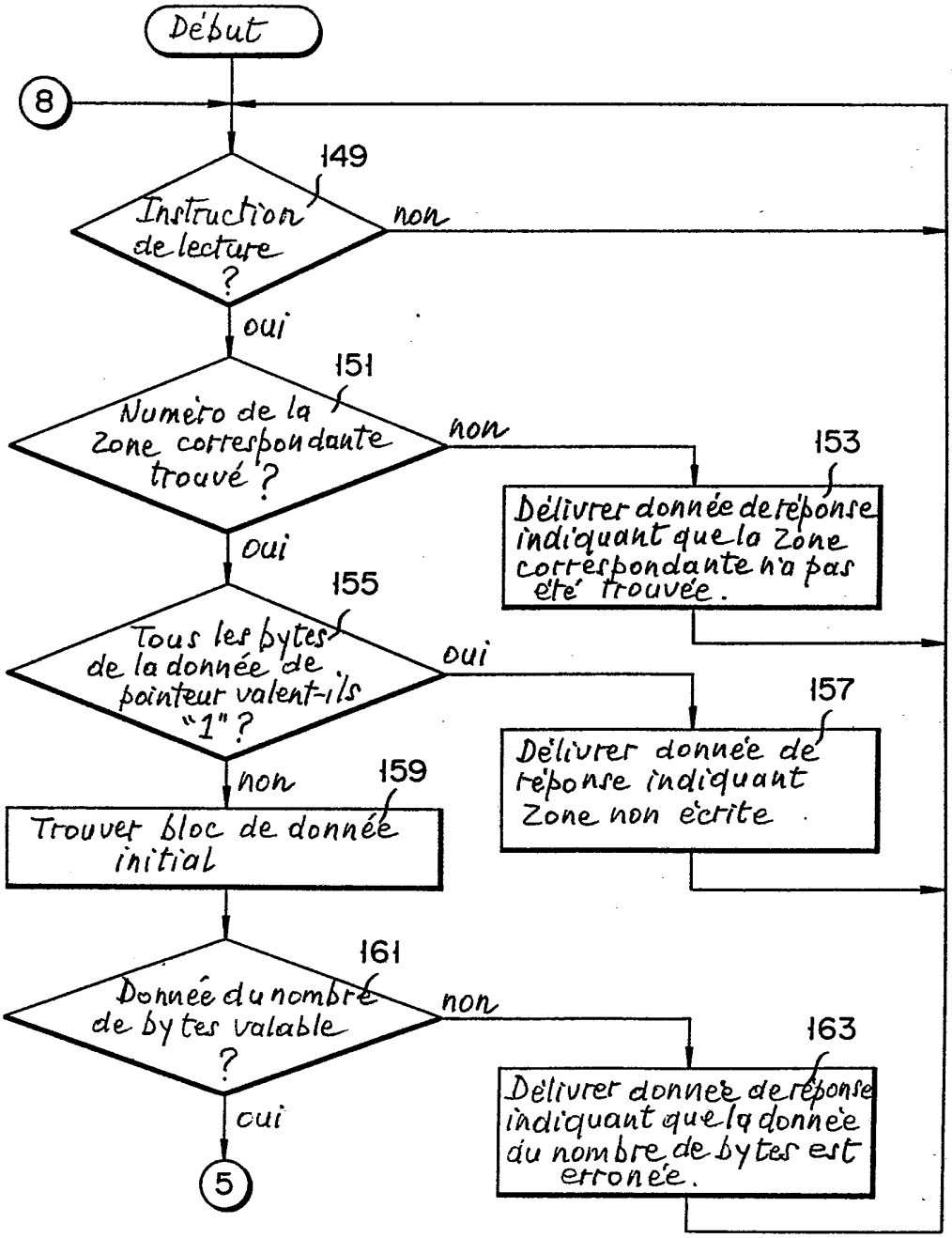

Début
8
149
Instruction de lecture ?
non
oui
151
Numéro de la zone correspondante trouvé ?
non
153
Délivrer donnée de réponse indiquant que la zone correspondante n'a pas été trouvée.
oui
155
Tous les bytes de la donnée de pointeur valent-ils "1" ?
oui
157
Délivrer donnée de réponse indiquant zone non écrite
non
159
Trouver bloc de donnée initial
161
Donnée du nombre de bytes valable ?
non
163
Délivrer donnée de réponse indiquant que la donnée du nombre de bytes est erronée.
oui
5

FIG. 18B

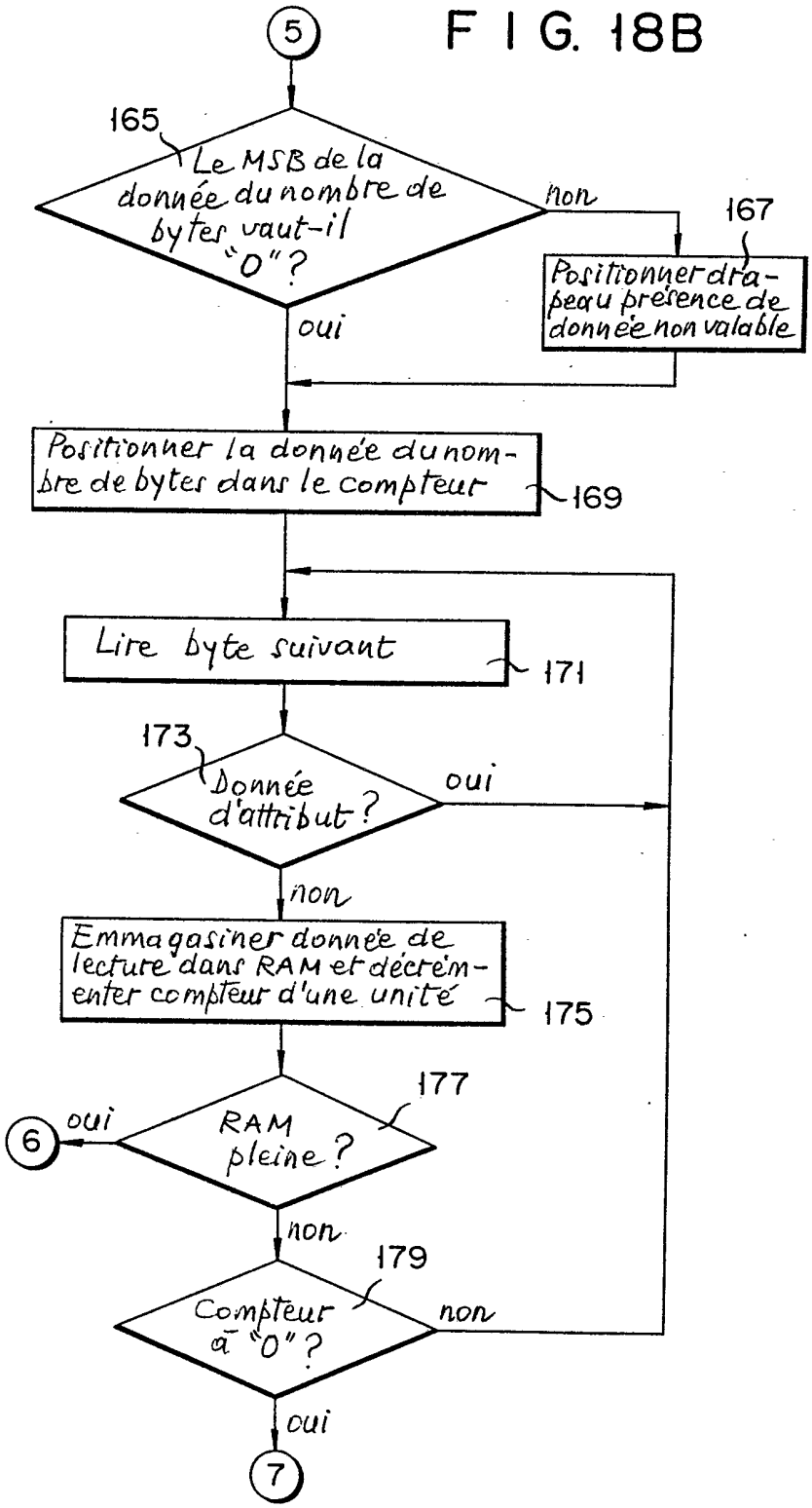

5
165
Le MSB de la donnée du nombre de bytes vaut-il "0"?
non
167
Positionner drapeau présence de donnée non valable
oui
Positionner la donnée du nombre de bytes dans le compteur
169
Lire byte suivant
171
173
Donnée d'attribut?
oui
non
Emmagasiner donnée de lecture dans RAM et décrémenter compteur d'une unité
175
177
RAM pleine?
oui
6
non
179
Compteur à "0"?
non
oui
7

FIG. 18C

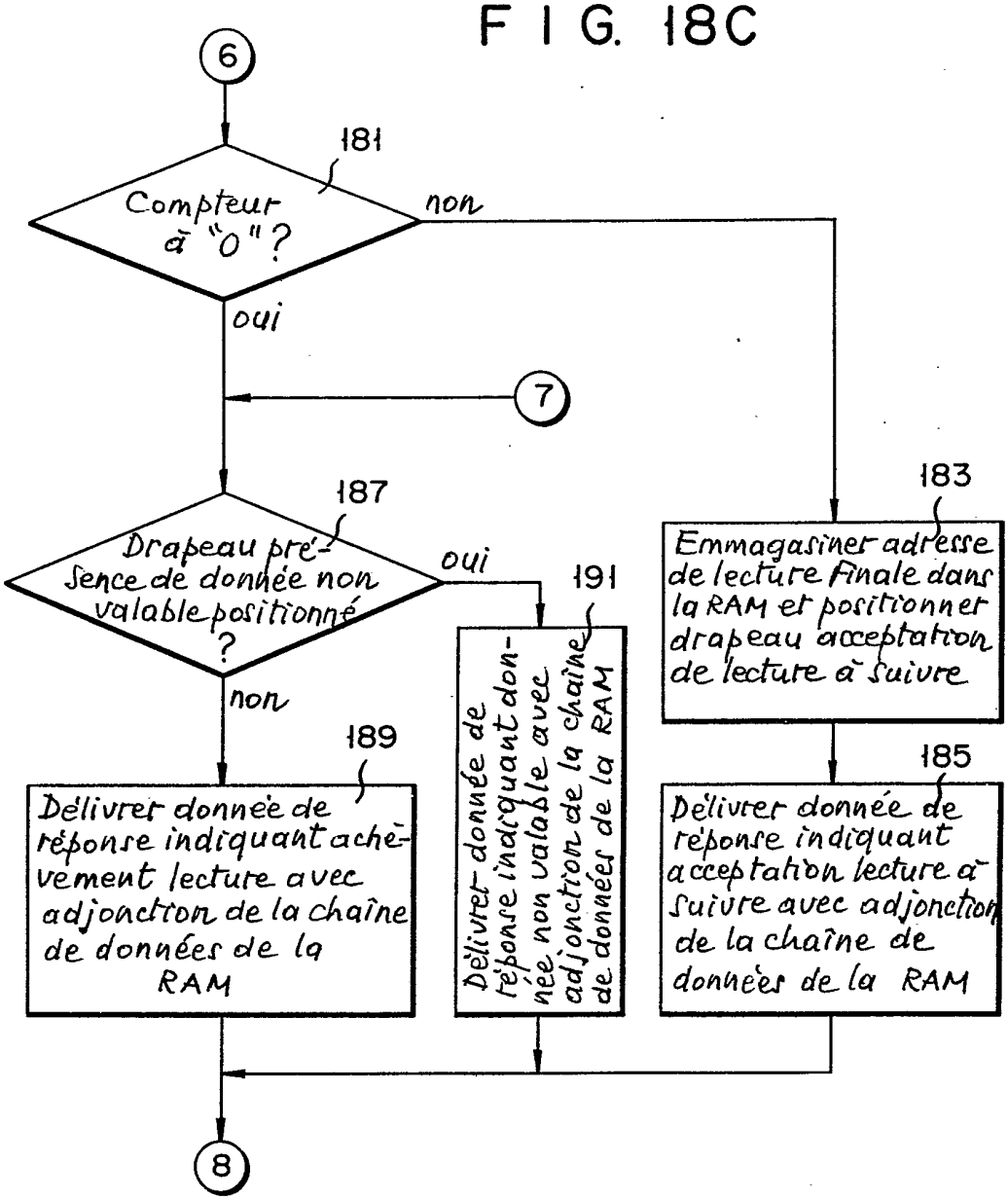


FIG. 19

Code de fonction de lecture	Numéro de zone

FIG. 20A

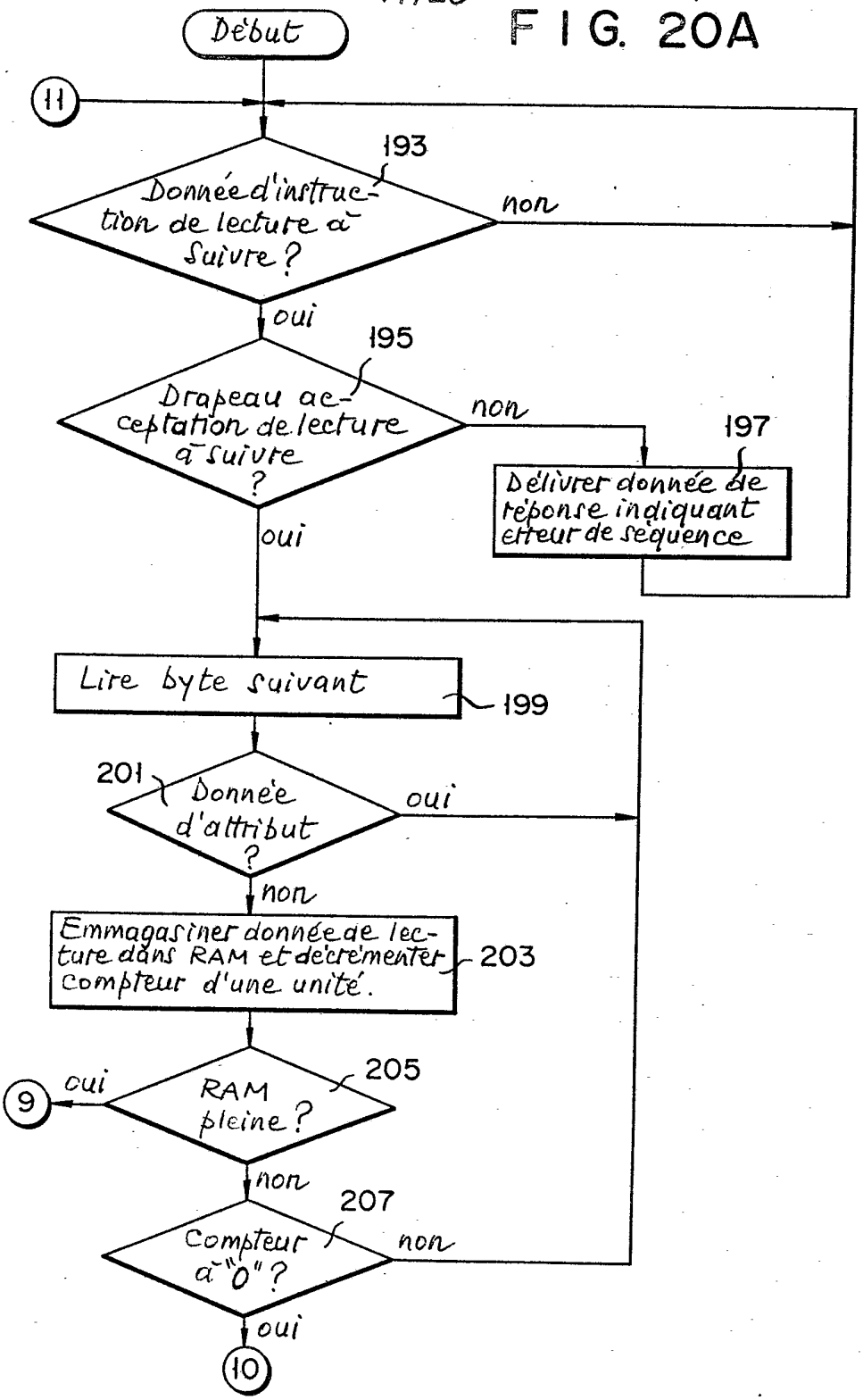

Début
11
193
Donnée d'instruction de lecture à suivre ?
non
oui
195
Drapeau acceptation de lecture à suivre ?
non
197
Délivrer donnée de réponse indiquant erreur de séquence
oui
Lire byte suivant
199
201
Donnée d'attribut ?
oui
non
Emmagasiner donnée de lecture dans RAM et décrémenter compteur d'une unité.
203
205
RAM pleine ?
oui
9
non
207
Compteur à "0" ?
non
oui
10

FIG. 20B

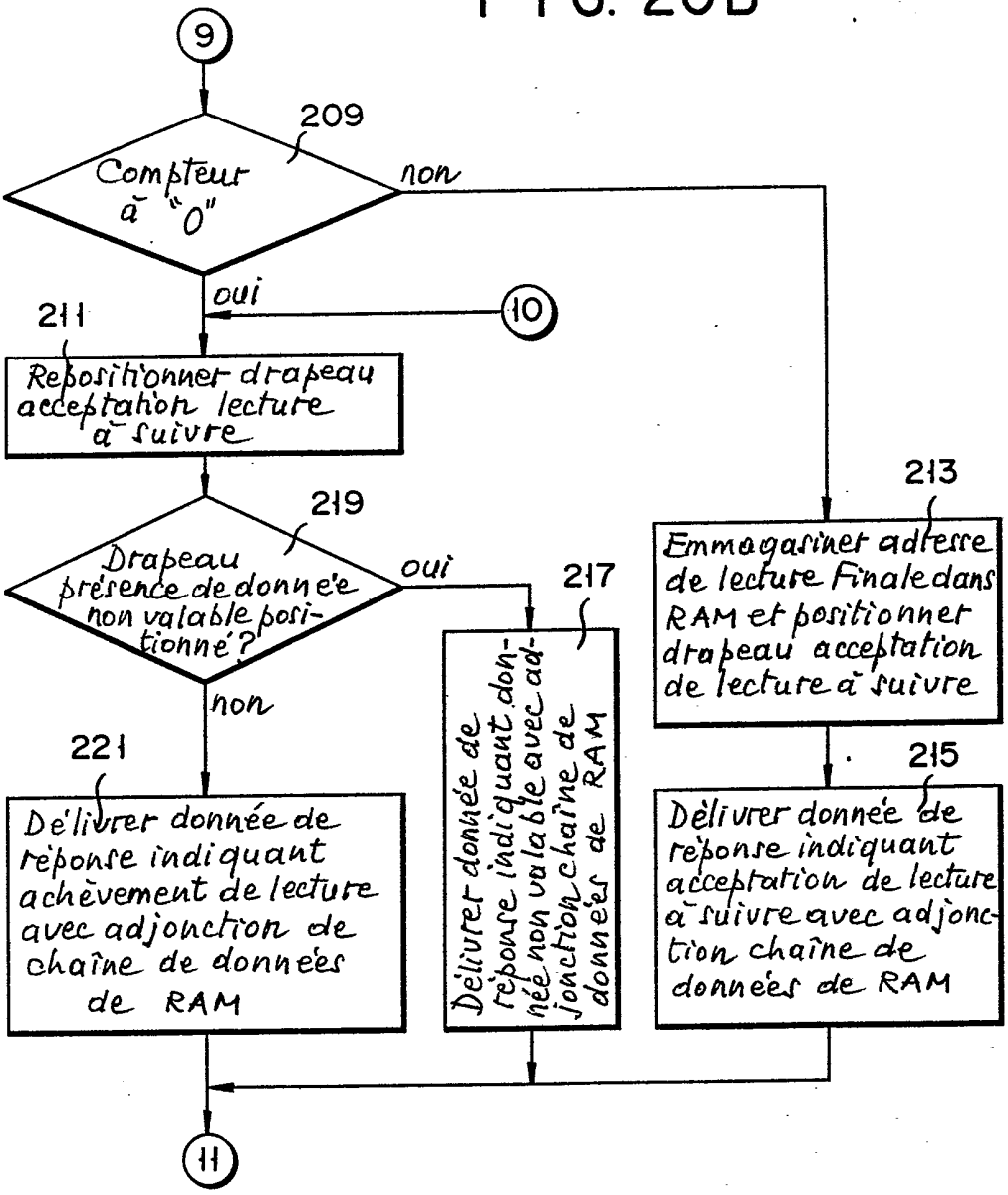


FIG. 20C

Code de Fonction de lecture à suivre	Numéro de zone

FIG. 21A

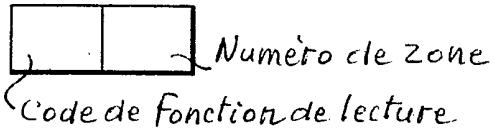


FIG. 21B



FIG. 21C

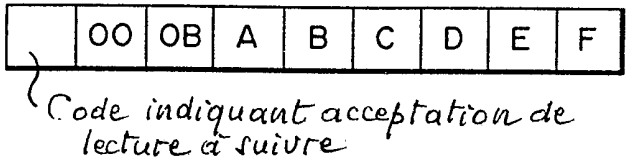


FIG. 21D

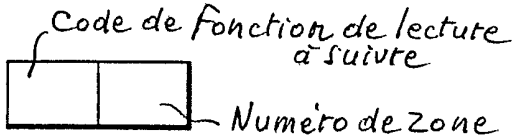
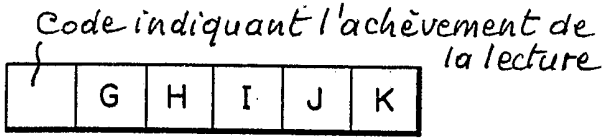


FIG. 21E



FIG. 21F



2591006

FIG. 22

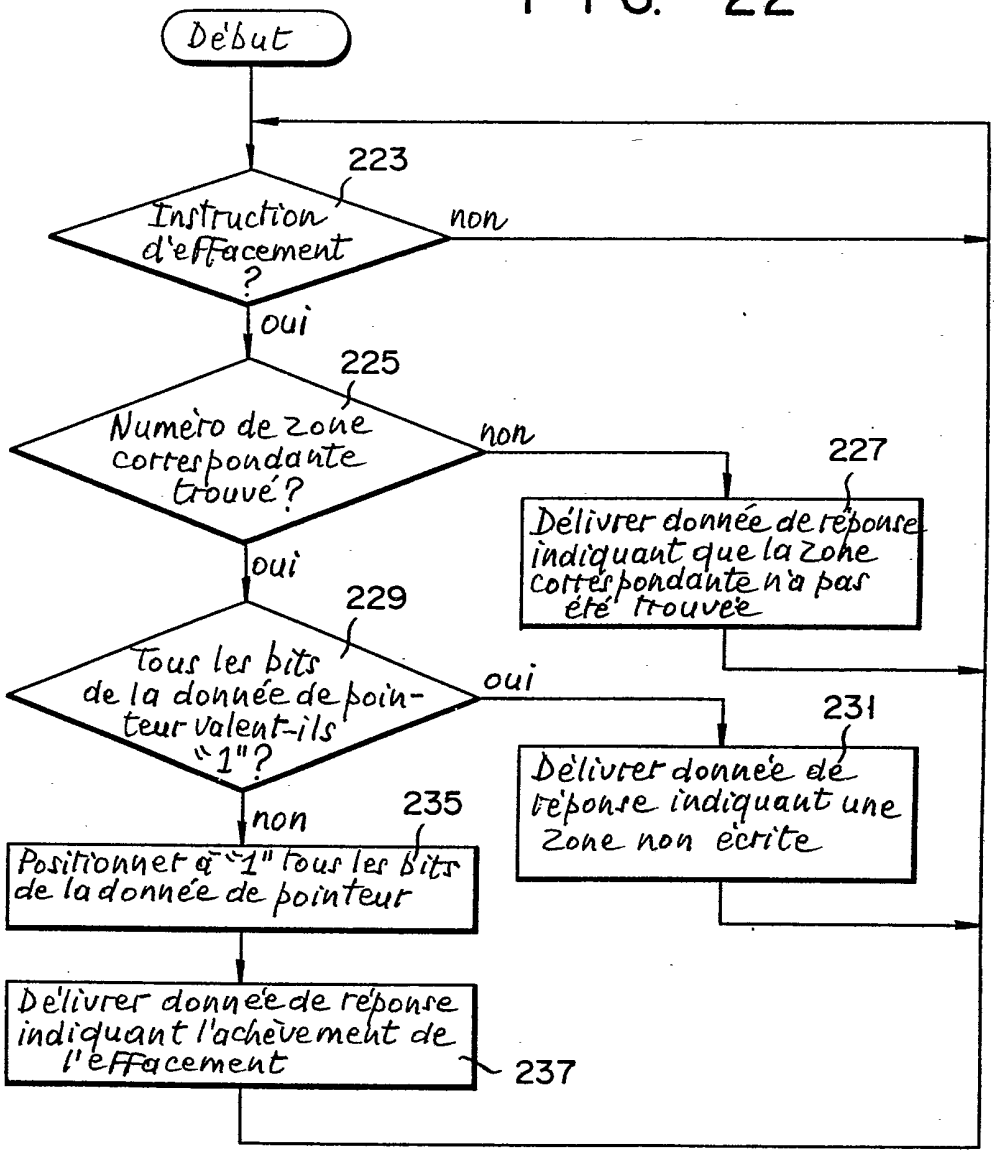


FIG. 23

Code de fonction d'effacement	Numéro de zone