

A1

**DEMANDE
DE BREVET D'INVENTION**

⑫

N° 81 17756

⑤④ Appareil de décodage pour des codes représentés par un arbre de code.

⑤① Classification internationale (Int. Cl.³). H 03 K 13/24.

②② Date de dépôt..... 21 septembre 1981.

③③ ③② ③① Priorité revendiquée : Japon, 22 septembre 1980, n° 131862/80.

④① Date de la mise à la disposition du
public de la demande..... B.O.P.I. — « Listes » n° 12 du 26-3-1982.

⑦① Déposant : Société dite : NIPPON TELEGRAPH & TELEPHONE PUBLIC CORPORATION, rési-
dant au Japon.

⑦② Invention de : Hideaki Kanayama.

⑦③ Titulaire : *Idem* ⑦①

⑦④ Mandataire : Cabinet Bonnet-Thirion et G. Foldés,
95, bd Beaumarchais, 75003 Paris.

La présente invention concerne un appareil permettant de décoder un code de longueur variable représenté par un réseau ou un arbre de code .

Un code connu de longueur variable représenté par
5 un arbre de code est par exemple le code de Huffman modifié. Ce code est obtenu par codage d'une longueur de parcours noire et blanche d'une image explorée, au moyen du système de codage de Huffman, comme cela est décrit par exemple dans l'article de R. Hunter et consorts "International Digital
10 Facsimile Coding Standards" (Norme internationale de codage de facsimilé numérique), Proceeding of the IEEE, juillet 1980, vol. 68, No. 7, pp. 854-867, chapitre IV "One-Dimensional Coding Scheme" (Système de codage unidimensionnel). Un procédé classique de décodage de ce code de Huffman modifié con-
15 siste à choisir l'une parmi deux branches en chaque noeud d'un arbre de code en fonction du fait que l'on détecte un "1" ou un "0" pour des bits individuels du code d'entrée, en aboutissant finalement au niveau d'un noeud final indiquant un résultat décodé. Pour réaliser ce décodage, il est néces-
20 saire d'accéder à une mémoire à tables de décodage pour chaque bit des données d'entrée et de décider, pour chaque sortie de la mémoire, si le noeud final est atteint ou non. Ceci augmente le nombre des phases opératoires de traitement, y compris l'accès à la mémoire, qui prend du temps, et le
25 traitement des décisions, ce qui entraîne que la durée globale de traitement devient relativement longue. C'est pour-quoi le procédé de décodage de l'art antérieur, mentionné ci-dessus, est désavantageux par le fait qu'on ne peut le mettre en oeuvre que moyennant l'utilisation d'un micropro-
30 cesseur à grande vitesse ou d'un circuit logique câblé.

Bien qu'il ne soit pas destiné à être utilisé pour le code de facsimilé, un procédé visant à accroître la cadence de décodage pour le code de Huffman est décrit dans le brevet déposé aux Etats-Unis d'Amérique sous le numéro
35 3.883.847, ayant pour titre "Uniform Decoding of Minimum-Redundancy Codes" (Décodage uniforme de codes à redondance minimale). Selon le procédé de décodage décrit dans ce brevet, les données d'un nombre prédéterminé de bits sont introduites à partir d'un train de codes d'entrée pour avoir accès

à une mémoire principale à tables de décodage en vue d'en effectuer la lecture et, dans le cas où le décodage n'est pas achevé au moyen de la lecture, une adresse d'une mémoire secondaire à tables de décodage, à laquelle on doit accéder
5 ensuite, est déterminée en utilisant les données extraites par lecture, puis des données d'entrée à plusieurs bits introduits ensuite ont alors accès à la mémoire secondaire à tables de décodage, ce qui permet d'obtenir un résultat décodé. Ce procédé accroît la vitesse de décodage, mais
10 utilise deux mémoires à tables de décodage et par conséquent présente l'inconvénient consistant en ce que la capacité globale de mémorisation est importante.

Un objet de la présente invention est de fournir un appareil de décodage pouvant réaliser un décodage à grande
15 vitesse d'un code de longueur variable représenté par un arbre de code, sans mettre en oeuvre l'opération de décodage bit à bit, et dont la mémoire à tables de décodage utilisée possède une faible capacité.

Conformément à un aspect de la présente invention,
20 la mémoire à tables de décodage possède une zone destinée à la mémorisation des données finales et une zone destinée à la mémorisation des données intermédiaires. Les données finales comportent au moins une information indiquant l'achèvement du décodage et une information indiquant le résultat
25 du décodage, et les données intermédiaires incluent au moins une information indiquant un décodage incomplet, une information indiquant le nombre des bits devant être introduits ensuite à partir d'un train de codes d'entrée et une information sur laquelle est basée l'adresse de la mémoire à tables
30 de décodage, à laquelle il faut ensuite accéder. Il se produit un accès à la mémoire à tables de décodage et une lecture de cette dernière, et des moyens de décision décident si les données lues hors de la mémoire à tables de décodage sont des données finales ou des données intermédiaires. Dans
35 le cas de données finales, l'information des données indiquant le résultat du décodage est délivrée par l'appareil de décodage. Dans le cas de données intermédiaires, les données ultérieures sont introduites à partir du train de codes d'entrée avec un nombre de bits qui est déterminé par l'informa-

tion des données intermédiaires indiquant le nombre des bits devant être introduits ultérieurement, et les données introduites et l'information des données intermédiaires pour la détermination de l'adresse de la mémoire à tables de décodage, à laquelle il faut ensuite accéder, sont traitées de manière à fournir l'adresse, à laquelle on doit accéder ultérieurement, puis l'accès à l'adresse de la mémoire à tables de décodage est ensuite réalisé. Ces opérations sont répétées jusqu'à ce que les données finales soient lues hors de la mémoire à tables de décodage.

Par détermination, à partir des données extraites par lecture, du nombre des bits du code devant être introduits ultérieurement et par détermination de l'adresse de la mémoire à tables de décodage, à laquelle il faut accéder ensuite, par exploitation des données introduites par un tel nombre prédéterminé de bits et de l'information extraite par lecture, il est possible de décoder à la fois un code possédant plusieurs bits sans qu'il soit nécessaire de sélectionner l'une de deux branches en chaque noeud de l'arbre de code, et il est possible de réduire le nombre de répétitions d'accès à la mémoire à tables de décodage, ce qui permet un décodage à grande vitesse. En outre on peut avoir un accès répété à la même mémoire à tables de décodage et par conséquent cette mémoire peut être utilisée de façon efficace et on peut lui donner une taille réduite. En outre lorsque le décodage d'un code est achevé, un premier bit du train de codes d'entrée indique le bit de tête du code suivant.

Conformément à un autre aspect de la présente invention, les données finales présentes dans la mémoire à tables de décodage incluent une information indiquant l'achèvement du décodage, une information indiquant le résultat du décodage et une information indiquant le nombre des bits devant être introduits ensuite, et les données intermédiaires incluent une information indiquant un décodage inachevé et une information sur laquelle est basée l'adresse de la mémoire à tables de décodage auxquelles on doit avoir accès ensuite. Après que la mémoire à tables de décodage ait été lue, des moyens de décision décident, en ce qui concerne les données extraites par lecture, s'il s'agit de données finales

ou de données intermédiaires. Dans le cas de données finales, l'information des données indiquant le résultat du décodage est sortie et les données ultérieures sont introduites à partir d'un train de codes d'entrée par le nombre de bits spécifiés par l'information des données extraites par lecture, indiquant le nombre de bits devant être introduits ensuite. Dans le cas de données intermédiaires, les données ultérieures sont introduites à partir du train de codes d'entrée par un nombre fixe m de bits (m étant un nombre entier supérieur ou égal à 2), et les données introduites et l'information des données extraites par lecture pour la détermination de l'adresse de la mémoire à tables de décodage, à laquelle doit s'effectuer ensuite l'accès, sont traitées, puis le résultat de l'opération est utilisé pour réaliser l'accès à la mémoire à tables de décodage. Ces opérations sont répétées pour le décodage successif des codes du train de codes d'entrée. Le nombre des bits devant être introduits ensuite dans les données finales est fourni par $L-(n-1).m$, où L est une longueur de code décodée et n est le nombre de répétitions d'accès à la mémoire à tables de décodage.

A titre d'exemple on a décrit ci-dessous et illustré schématiquement aux dessins annexés une forme de réalisation de l'objet de l'invention.

La figure 1 est un diagramme montrant une partie d'une table de décodage pour le code de Huffman modifié, au moyen d'un procédé de décodage bit par bit utilisant un arbre de code.

La figure 2 est un diagramme montrant les phases opératoires du décodage d'un code d'entrée "11011" indicatif d'une longueur de parcours blanche 64, grâce à l'utilisation de la table de décodage représentée sur la figure 1.

La figure 3 est un schéma-bloc illustrant un appareil de décodage décrit dans le brevet déposé aux Etats-Unis d'Amérique sous le No. 3.883.847.

La figure 4 (4A, 4B) est un diagramme montrant un exemple d'une table de décodage pour des codes blancs, qui est utilisée dans l'appareil de décodage conforme à la présente invention.

La figure 5A montre un exemple du format de données

finales.

La figure 5B montre un exemple du format de données intermédiaires.

La figure 6 est un diagramme montrant les phases opératoires du décodage d'un code d'entrée "11011" indicatif d'une longueur de parcours blanche 64, moyennant l'utilisation de la table de décodage représentée sur la figure 4 (4A, 4B).

La figure 7 montre un schéma-bloc d'un exemple de l'appareil de décodage conforme à la présente invention, utilisant la table de décodage de la figure 4 (4A, 4B).

La figure 8 montre un exemple d'une partie de la table de décodage pour des codes blancs, destinée à être utilisée dans une autre forme de réalisation de l'appareil de décodage conforme à l'invention.

La figure 9 est un diagramme montrant les phases opératoires de décodage d'un code d'entrée "11011" représentatif d'une longueur de parcours blanche 64, moyennant l'utilisation de la table de décodage de la figure 8.

La figure 10 montre un schéma-bloc représentatif d'une autre forme de réalisation de l'appareil de décodage conforme à la présente invention utilisant la table de décodage de la figure 8.

La figure 11 montre un schéma-bloc représentatif d'une autre forme de réalisation de l'appareil de décodage conforme à la présente invention, utilisant la table de décodage de la figure 4 (4A, 4B) et construit en utilisant un microprocesseur.

La figure 12 est un organigramme montrant un exemple de fonctionnement de l'appareil de décodage représenté sur la figure 11.

La figure 13, (13A, 13B) est un diagramme, correspondant à celui de la figure 4 (4A, 4B), montrant un exemple d'une table de décodage pour un code noir.

La figure 14 est un graphe montrant les relations entre différents systèmes de décodage et les capacités de mémoire à tables de décodage requises pour ces systèmes.

La figure 15 est une table montrant les relations entre différents systèmes de décodage et les temps requis

pour accéder aux mémoires à tables de décodage.

La figure 16 est une table montrant les relations entre les capacités des mémoires à tables de décodage et les durées d'accès requises dans chaque appareil de décodage, dans le cas où la capacité de la mémoire à tables de décodage et les temps d'accès aux mémoires requises dans la première forme de réalisation de la présente invention sont représentées chacune par 1.

Afin de faciliter une meilleure compréhension de la présente invention, on va fournir tout d'abord une description du procédé classique de décodage bit par bit. La figure 1 représente une partie d'un arbre de code blanc du code de Huffman modifié, basé sur le procédé classique de décodage bit par bit, l'arbre de code constituant une table de décodage. Une racine 1 est subdivisée en deux branches 4, dont chacune atteint un noeud 2. Le noeud 2 se subdivise en deux branches 4, dont chacune est à nouveau subdivisée en deux branches 4 en chaque noeud 2. De cette manière, on atteint les noeuds finaux 3. Au-dessus de la branche 4, au niveau de chaque noeud 2 et avant que ce dernier ne soit subdivisé, on a représenté une adresse <XX> de la table de décodage et des données (XX) de la table de décodage ou une longueur parcourue [XX]. Les X de cette adresse et de ces données prennent la forme de l'un quelconque des caractères 0, 1, 2, 3, ... 9, A, B, C, D, E et F ; en effet "XX" représente des nombres hexadécimaux à deux chiffres. Cependant, dans le cas de la longueur de parcours, le X indique un nombre décimal. Une lettre minuscule "c" indique une extension supplémentaire de l'arbre de code .

La figure 2 représente un exemple du décodage d'un code "11011" représentant une longueur de parcours blanche 64, qui est introduite en tant que train de codes d'entrée (train de données) dans la table de décodage de la figure 1. La procédure de décodage commence avec la préparation de (00) comme données initiales. Le premier bit "1" des données d'entrée est introduit en étant ajouté aux données initiales (00) de manière à obtenir une adresse de saut <01>. Des données (04) de l'adresse de saut sont extraites par lecture. A partir des données (04), il est vérifié si un noeud final

de l'arbre de codes a été atteint ou non. Les données ($X_2=0$, $X_1=4$) sont représentées respectivement sous la forme de nombres binaires et le fait qu'un noeud final a été atteint ou non est décidé en fonction du fait que le bit de poids le plus important de " X_2X_1 " c'est-à-dire "0000100" est "1" ou "0". Dans ce cas, étant donné que le bit de poids le plus important est "0", il est décidé que le noeud final n'a pas été atteint (dans le cas où le bit de poids le plus important est "1", il est décidé que le noeud final a été atteint). Ensuite le bit suivant "1" des données d'entrée est ajouté aux données (04) de manière à obtenir une adresse de saut <05>. De cette manière l'une des deux branches est choisie pour chaque noeud 2 en fonction du fait que chaque donnée à un bit est "0" ou "1". Ensuite, l'accès à la table de décodage, la décision du noeud final et la formation d'une adresse de saut sont répétés de façon similaire lors de chaque application d'un bit des données d'entrée, ce qui permet d'arriver en définitive au niveau du noeud final.

Ainsi une adresse <21> est atteinte et sa donnée (C1), c'est-à-dire "11000001" est lue. Etant donné que le bit de poids le plus important de cette donnée est "1", il est décidé que le noeud final a été atteint, et la longueur de parcours est décodée en utilisant la donnée (C1) de ce noeud final. C'est-à-dire que la donnée du bit qui succède au bit de poids le plus important de la donnée (C1=11000001) indique un code de terminaison ou un code de reprise. Lorsque où ce bit est "1", cela indique le code de reprise et la longueur de parcours est obtenue en multipliant 64 par un nombre formé des six chiffres d'ordre inférieur, c'est-à-dire "000001" (un nombre binaire) = 1 (un nombre décimal) dans ce cas. Lorsque la donnée du bit d'ordre immédiatement inférieur par rapport au bit de poids le plus important des données lues est "0", cela indique le code de terminaison et la valeur des six bits d'ordre inférieur de la donnée indique la longueur de parcours. Dans l'art antérieur, il était nécessaire d'accéder à la table de décodage pour chaque bit de donnée et de prendre une décision relative à la donnée, comme cela a été décrit ci-dessus. Ceci entraîne de façon inévitable une réduction de la vitesse de décodage.

En se référant à la figure 3, on va donner une brève description d'un procédé de décodage décrit dans le brevet déposé aux Etats-Unis d'Amérique sous le No. 3.883.847. Les données d'entrée sont introduites dans un registre d'entrée 5 210, à partir duquel seulement K bits des données d'entrée sont transférés dans un registre 211. Les données à K bits ainsi appliquées au registre 211 sont incrémentées de 1 dans un additionneur 212, dont le signal de sortie est utilisé en tant qu'adresse pour l'accès à une mémoire à tables de décodage principale 214 par l'intermédiaire d'un circuit d'adres- 10 sage 213 et d'une porte d'inhibition 281. Lorsque les données provenant de la mémoire 214 sont un résultat décodé, le code décodé est délivré à partir d'un registre 217 à un conducteur de sortie 243 par l'intermédiaire d'une porte 15 d'inhibition 241 et d'une porte OU 242. En outre le nombre des bits utilisés dans les données à K bits introduites est délivré par un registre 216 à un multiplicateur de cadence binaire 218 par l'intermédiaire d'une porte d'inhibition 283 et d'une porte OU 286. Le registre d'entrée 210 est commandé 20 par le multiplicateur de cadence binaire 218 conformément au nombre des bits utilisés de ce fait pour introduite la quantité correspondante des données d'entrée dans le registre 211.

D'autre part, dans le cas où les K bits des données 25 introduites dans le registre 211 ne constituent pas un code, le registre 217 délivre un signal indiquant qu'il n'y a aucune délivrance d'un résultat décodé lorsqu'intervient l'accès à la mémoire à tables principale. Ce signal provoque l'inhibition de la porte d'inhibition 283 et le positionne- 30 ment d'une bascule bistable 285. Le signal de sortie de la bascule bistable 285 est envoyé aux portes d'inhibition 241 et 281 en vue de les inhiber. En outre le signal de sortie provenant du registre 217 est appliqué par l'intermédiaire de la porte 286 au multiplicateur 218 afin de piloter ce 35 dernier, en introduisant les K bits des données d'entrée dans le registre 211 par l'intermédiaire du registre 210. Des bits inutiles dans les données du registre 211 sont masqués par le signal de sortie délivré du registre 216 dans un décodeur 260 de manière à retirer les bits A requis. Le

signal de sortie provenant du décodeur 260 est incrémenté de 1 dans un additionneur 261 et le résultat additionné est utilisé en tant qu'adresse pour accéder à une mémoire à tables secondaire 250 par l'intermédiaire d'un circuit d'adressage 251. Dans ce cas le circuit d'adressage 251 est commandé par le signal de sortie provenant du registre 217 de manière à déterminer à laquelle des tables de la mémoire secondaire à tables 250 on a accès. Lorsque la mémoire 250 est lue, le résultat décodé est délivré à partir d'un registre 272 au conducteur de sortie 243 par l'intermédiaire de portes 292 et 242 et le nombre des bits utilisés pour le décodage est envoyé depuis un registre 271 au multiplicateur 218 par l'intermédiaire de la porte 286. Lors de la détection de la fin des données d'entrée par un détecteur 221, une bascule bistable 222 est positionnée et son signal de sortie est envoyé à une porte d'inhibition 224 de manière à fermer cette dernière, ce qui arrête le fonctionnement de cet appareil de décodage.

L'appareil de décodage conforme au brevet déposé aux Etats-Unis d'Amérique mentionné précédemment est du type qui réalise un accès à la mémoire principale à tables 214 utilisant plusieurs bits des données d'entrée et dans lequel il n'y a pas de vérification pour savoir si chaque bit est décodé ou non ; par conséquent le nombre des accès requis à la mémoire est faible, ce qui permet un décodage à grande vitesse. Cependant cet appareil nécessite de prévoir les première et seconde mémoires à tables 214 et 250, ce qui entraîne une taille accrue de mémoire.

La figure 4 (4A, 4B) illustre un exemple d'une table de décodage destinée à être utilisée dans l'appareil de décodage conforme à la présente invention ; il s'agit d'une table de décodage utilisant un arbre de code blanc du code de Huffman modifié. Dans ce cas la longueur des données (XX) est de huit bits, l'adresse de mémoire <XX> est de huit bits et le nombre de branches devant être sélectionnées à la fois, c'est-à-dire le nombre des bits de code devant être décodés à la fois, est 1, 2 ou 4. Il va de soi que la longueur des données, la longueur de l'adresse de mémoire et le nombre des branches devant être sélectionnées à la fois

sont arbitraires. De même le nombre des branches devant être sélectionnées à la fois peut être fixé ou limité de façon spécifique à par exemple deux. Les chiffres de référence 1 à 4 sont identiques à ceux utilisés sur la figure 1, 5 mais une donnée pendant le cours du décodage est indiquée par un noeud interne 5. Ici le nombre des branches devant être choisi à la fois est sélectionné de manière à ne pas dépasser le nombre de branches 4 depuis la racine 1 ou le noeud interne 5 jusqu'au noeud final 3 le plus proche. En 10 effet dans la figure 4 (4A, 4B) les noeuds finaux 3 les plus proches par rapport à la racine 1 sont ceux possédant les adresses <07>, <08>, <0B>, <0C>, <0E> et <0F> sur une ligne 61 et ce sont les quatrièmes à partir de la racine 1. En examinant les noeuds internes sur la ligne 61 par exemple, 15 une adresse <03> atteint le noeud final 3 le plus proche au niveau d'une adresse <10> qui est une branche proche de l'adresse <03> et le nombre des branches choisi est égal à un. Cependant dans le cas d'une adresse <00>, il n'existe aucun noeud final avant une seconde branche, c'est-à-dire 20 une branche située sur la ligne 62, par exemple lorsqu'une adresse <13> est atteinte. C'est pourquoi le nombre des branches choisi à partir de l'adresse <00> jusqu'au noeud final le plus proche est deux. Les données (FF) et (EF) indiquent un signal de synchronisation de ligne EOL et les 25 données (8F) indiquent une erreur de décodage.

Les données (XX) sont mémorisées dans chaque adresse <XX> d'une mémoire à tables de décodage succédant à la table de décodage de la figure 4 (4A, 4B). [XX] indique la longueur de parcours et alors X est un nombre décimal. La mémoire 30 de décodage possède une zone destinée à la mémorisation des données finales et une zone destinée à la mémorisation des données intermédiaires. Dans cette forme de réalisation la mémoire à tables de décodage est une mémoire à un mot à huit bits et les données finales et les données intermédiaires 35 sont chacune formée de huit bits. Le format des données finales est tel que représenté sur la figure 5A. Lorsque le bit de poids le plus faible B_1 est "1", cela indique un noeud final 3, c'est-à-dire une donnée finale. Le bit B_2 succédant au bit de poids le plus faible B_1 indique le type

du code décodé ; dans cet exemple, il indique si le résultat décodé est un code de terminaison ou un code de reprise.

Les six bits B_3 à B_8 sont des informations indiquant le résultat décodé, c'est-à-dire la longueur parcourue. Lorsque le bit B_2 est égal à "1", l'information du résultat décodé est le code de terminaison et, comme cela est indiqué par le chiffre de référence 44, les bits B_3 à B_8 possèdent les poids respectifs 2^0 à 2^5 . Lorsque le bit B_2 est "0", l'information du résultat décodé est le code de reprise et, comme cela est indiqué par le chiffre de référence 45, les bits B_3 à B_8 ont pour poids respectifs 2^6 jusqu'à 2^{11} .

Le format des données intermédiaires est tel que représenté sur la figure 5B. Lorsque le bit de poids le plus faible B_1 est "0", il représente le noeud interne 5, c'est-à-dire qu'il indique que la donnée est une donnée intermédiaire dans le cours du décodage. Le second bit B_2 indique le nombre des bits devant être introduits ensuite à partir du train de codes d'entrée. Lorsque le bit B_2 est "0", un bit est introduit et lorsque le bit B_2 est "1", deux bits sont introduits. Les bits B_3 à B_8 représentent une information pour déterminer l'adresse de la mémoire à tables de décodage, à laquelle l'accès doit être ensuite effectué.

La figure 6 montre un exemple de décodage dans le cas où "11011" indiquant la longueur de parcours blanche 64 est introduite en tant que donnée d'entrée dans la table de décodage de la figure 4 (4A, 4B). L'opération de décodage de ce cas va être décrite en référence à la figure 6. (00) est préparé en tant que donnée initiale. Comme cela a été décrit précédemment, le nombre des branches devant être sélectionnées tout d'abord, c'est-à-dire le nombre des bits devant être introduits au départ du décodage de chaque code, est quatre et quatre bits "1101" ("OD" dans le système hexadécimal) de la donnée d'entrée sont ajoutés à la donnée initiale (00) en vue d'obtenir l'adresse de saut <OD>. A partir de l'adresse de saut se trouve lue sa donnée (34). Par détection du bit de poids le plus faible B_1 de la représentation binaire "00110100" de cette donnée, il est décidé si un noeud final 3 de l'arbre de code a été atteint ou non. Etant donné que le bit de poids le plus faible B_1 est

"0", cette donnée est une donnée intermédiaire et son décodage n'a pas été achevé. A partir du second bit $B_2=0$ de la représentation binaire de la donnée lue (34), il est décidé que le nombre des branches devant être sélectionnées ensuite est un. Ensuite le bit "1" suivant de la donnée d'entrée est introduit et il est ajouté à une donnée, dans laquelle les bits B_1 et B_2 des données lues (34) ont été rendus égaux à "0", ce qui permet d'obtenir une adresse de saut <35>. A partir de l'adresse de saut se trouve lue sa donnée (07).

10 Etant donné que le bit de poids le plus faible B_1 de la représentation binaire "00000111" de la donnée est "1", il est décidé qu'un noeud final de l'arbre de code a été atteint. A partir du bit $B_2=1$ de la donnée lue, il est décidé que la donnée est le code de reprise. Ainsi l'information du résultat codé $B_8B_7 \dots B_3=000001$ signifie une longueur de parcours égale à 64, qui est obtenue par multiplication du résultat décodé ("1" en nombres décimaux) par 64.

La figure 7 illustre une forme de réalisation de l'appareil de décodage de la présente invention, basée sur l'exemple de la table de décodage du code de Huffman modifié représenté sur la figure 4 (4A, 4B). Le fonctionnement de cet appareil va être décrit ci-après en liaison avec le cas où la donnée d'entrée "11011" est introduite comme dans le cas de la figure 6. Pour le démarrage de l'opération de décodage, un signal de démarrage de décodage est délivré sous la forme d'un niveau élevé "1" à partir d'un conducteur 11 à un circuit 12 de détection d'achèvement de décodage, par exemple pendant un certain intervalle de temps. Par suite de la montée du signal de sortie provenant d'une ligne 25 32 délivrant le signal de sortie du circuit de détection 12, une bascule bistable 14, indiquant si la donnée d'entrée est une donnée blanche ou une donnée noire, est ramenée à l'état initial de manière à préparer le décodage d'un code blanc et, simultanément, une bascule bistable d'adresses de mémoire 15 est ramenée à l'état initial. Pour le code de Huffman modifié, l'opération de décodage d'un code de facsimilé commence avec un code blanc. Une adresse de mémoire provenant de la bascule bistable 15 est utilisée pour effectuer la lecture d'une mémoire 16, dans laquelle se trouve

mémorisée la table de décodage de la figure 4 (4A, 4B), et la donnée initiale (00) extraite par lecture est délivrée à une bascule bistable de données 17. Dans le circuit 12 de détection de l'achèvement du décodage, le signal mentionné

5 ci-dessus de départ du décodage fournit un niveau bas dans une ligne de sortie 18 par l'intermédiaire d'une porte OU 33 et d'un inverseur 34. Simultanément le signal de sortie provenant de l'inverseur 34 produit un niveau bas dans une ligne de sortie 38 par l'intermédiaire d'un circuit 39

10 d'allongement des impulsions, constitué par une ligne à retard et une porte ET. Des sélecteurs de données 19, 20 et 21 sont respectivement raccordés au côté b, au moyen de ces niveaux. Sous l'effet du fonctionnement indiqué ci-dessus, le sélecteur de données 20 est raccordé à une mémoire de

15 données initiales 23, aussitôt après le démarrage de l'opération de décodage. La mémoire de données initiales 23 est commandée par un signal de sortie "0" (indiquant un décodage de code blanc) provenant de la bascule bistable 14 de manière à régler une valeur 4 dans un compteur de données d'entrée

20 24 par l'intermédiaire du sélecteur 20. Les signaux d'horloge d'entrée provenant d'une borne 25 sont appliqués au compteur 24 et, en synchronisme avec ces signaux d'horloge, une donnée d'entrée à quatre bits "1101" est introduite dans un convertisseur série-parallèle des données d'entrée,

25 à partir d'un conducteur 27. Ensuite la donnée d'entrée "1101" est envoyée par le convertisseur série-parallèle 26 à un circuit additionneur à huit bits 22, après quoi le convertisseur série-parallèle 26 des données d'entrée est ramené à l'état initial. C'est-à-dire que son état d'entrée est

30 rendu égal à "0000", tandis que ses sorties sont maintenues non modifiées, c'est-à-dire restent "1101". Pendant ce temps, le bit de poids le plus faible $B_1=0$ de la donnée initiale (00) introduite dans la bascule bistable des données 17 comme décrit précédemment est appliquée au circuit 12 de

35 détection de fin de décodage de manière à fournir un niveau élevé dans la ligne de sortie 18 par l'intermédiaire de la porte OU 33, ce qui provoque une commutation des sélecteurs 19 et 21 sur le côté a. Il en résulte que les bits B_8 à B_3 de la donnée initiale (00) sont appliqués par la bascule

17 à l'additionneur à huit bits 22 par l'intermédiaire du sélecteur 21. Cette donnée initiale (00) et la donnée d'entrée "1101" provenant du convertisseur série-parallèle 26 sont additionnées dans le circuit additionneur à huit bits 22 de manière à fournir une adresse de saut <OD>. Le sélecteur 20 est commuté sur le côté a par le circuit 39 d'allongement des impulsions après la commutation des sélecteurs 19 et 20, c'est-à-dire après la sortie de l'adresse de saut <OD>.

L'adresse de saut <OD> est délivrée à la bascule bistable d'adresses de mémoire 15. Le signal de lecture présent dans une ligne de transmission de signaux 28 a accès à la mémoire 16 en vue de lire la donnée (34) correspondant à l'adresse de saut <OD>, qui est envoyée à la bascule bistable de données 17. Par conséquent, dans ce cas, $B_1=0$ et $B_2=0$. Etant donné que le bit de poids le plus faible B_1 est "0", cette donnée n'est pas une donnée finale, mais une donnée intermédiaire. Par conséquent les sélecteurs de données 19 à 21 sont maintenus sur le côté a par le circuit 12 de détection de fin de décodage. Le bit $B_2=0$ est délivré en tant qu'adresse à une mémoire suivante de données 36 pour lire le contenu "1" qui à son tour est comptabilisé dans le compteur 24 des données d'entrée par l'intermédiaire du sélecteur 20. Par conséquent le bit "1" suivant de la donnée d'entrée provenant du conducteur 27 est introduit dans le convertisseur série-parallèle des données d'entrée 26 en synchronisme avec le signal d'horloge d'entrée. Le signal de sortie du convertisseur série-parallèle des données d'entrée 26 passe à l'état "0001" et le signal d'entrée au circuit additionneur à huit bits 22 passe à l'état (01). Les bits B_8 à B_3 du signal de sortie (34) provenant de la bascule bistable des données 17 sont appliqués au circuit additionneur à huit bits 22. (01) et (34) - " B_2B_1 " sont additionnés par le circuit additionneur à huit bits 22 pour fournir une adresse de saut <35>, qui est envoyée à la bascule bistable d'adresses de mémoire 15. A partir de l'adresse <35> de la mémoire 16 se trouve lue une donnée (07) qui est envoyée à la bascule bistable des données 17. Par conséquent les bits B_1 et B_2 passent tous les deux à l'état "1".

Etant donné que le bit de poids le plus faible B_1

est "1", le circuit 12 de détection de fin de décodage détecte que la donnée est une donnée finale, c'est-à-dire qu'un noeud final a été atteint, et le signal de sortie détecté entraîne une commutation des sélecteurs de données 5 19 à 21 sur le côté b. Etant donné que le bit B_2 est "1", un sélecteur de données 29 est commuté sur le côté MC et les bits B_8 à B_3 présents dans la bascule bistable de données 17 sont appliqués aux six bits d'ordre supérieur d'une bascule bistable de délivrance de la longueur de parcours. Il en résulte que la valeur "000001000000", c'est-à-dire la longueur de 10 parcours "64" est délivrée, ce qui termine l'opération de décodage. Dans le code de Huffman modifié, lorsqu'une longueur de parcours est supérieure à 63, cette longueur est représentée par une combinaison d'un code de reprise et 15 d'un code de terminaison. Par exemple un parcours blanc 80 est représenté par un parcours blanc 64 + un parcours blanc 16. Par conséquent le décodage pour un code de reprise est toujours suivi par un code de terminaison de la même couleur que le code précédent, ce qui est accompli dans l'appareil 20 de décodage conforme à la présente invention de telle manière que le résultat décodé du code de reprise, c'est-à-dire la donnée finale, fournit toujours "1" comme bit suivant immédiatement le bit de poids le plus faible B_1 , et le bit $B_2=1$ de la donnée finale est envoyé au circuit 12 de détection de fin de décodage afin d'inhiber sa porte d'inhibition 25 35. C'est pourquoi le bit de poids le plus faible $B_1=1$ est envoyé à la porte 33 et que l'opération de décodage pour le code suivant commence avec le code blanc. Dans le cas du parcours blanc 64, comme explicité précédemment, la donnée est 30 décodée sous la forme d'un parcours blanc 64 + parcours blanc 0. Dans le cas où la donnée décodée est un code de terminaison ou de fin, le bit B_2 de la donnée finale est "0" et le bit $B_1=1$ est délivré par l'intermédiaire de la porte d'inhibition 35 à la bascule bistable 14 afin d'inverser 35 cette dernière, en modifiant la lecture de la mémoire à tables de décodage 16 depuis la table de décodage des codes blancs à une table de décodage des codes noirs et réciproquement. Simultanément la bascule bistable d'adresses de mémoire 15 est ramenée à l'état initial de manière à décoder la

donnée d'entrée suivante de la même manière que dans le cas de la donnée d'entrée "11011".

La figure 8 montre une partie de la table de décodage du code de Huffman modifié dans une autre forme de réalisation de l'appareil de décodage conforme à la présente invention. Sur la figure 8 les parties correspondant à celles de la figure 4 (4A, 4B) sont repérées par les mêmes chiffres de référence. Dans ce cas la longueur des données est de 10 bits, la mémoire d'adresses est de huit bits et le nombre des branches devant être sélectionnées à la fois, c'est-à-dire le nombre des bits d'une donnée d'entrée devant être introduits ultérieurement dans le cas du décodage non terminé, est fixé à quatre. De même dans ce cas, X est un chiffre d'un nombre hexadécimal et si l'affectation des bits dans le cas de chaque chiffre de la donnée (XXX), qui est converti sous la forme de la représentation binaire, est choisie comme étant $(B_{10}B_8B_7B_6B_5XXB_4B_3B_2B_1)$, alors le bit B_1 indique la fin ou la continuation de l'opération de décodage ; les bits B_2 et B_3 indiquent le nombre des bits de données initiales devant être introduits après la fin de l'opération de décodage, c'est-à-dire lorsque la donnée extraite par lecture est une donnée finale ; le bit B_4 indique si la donnée décodée est un code de reprise ou de terminaison à la fin de l'opération de décodage ; et les bits B_5 à B_{10} indiquent une donnée d'adresse dans le cas de la donnée intermédiaire et le résultat décodé dans le cas de la donnée finale. "X" est "1" ou "0" et ceci n'est pas utilisé pour l'opération de décodage.

La figure 9 montre un exemple du déroulement du décodage dans le cas où la table de décodage de la figure 8 est utilisée et où la donnée "11011" indiquant la longueur de parcours blanc 64 est introduite en tant que donnée d'entrée. Ce fonctionnement sera décrit plus loin.

La figure 10 montre une autre forme de réalisation de l'appareil de décodage selon la présente invention, qui utilise la table de décodage du code de Huffman modifié, représentée sur la figure 8. Sur la figure 10 les parties correspondant à celles de la figure 7 sont repérées par les mêmes chiffres de référence. Le sélecteur de données

20 choisit si la valeur du compteur de données d'entrée 24 est réglée à un nombre fixe de la donnée d'entrée à partir d'une mémoire 35 de réglage d'un nombre fixe de bits ou au nombre des données provenant d'un circuit 37 de réglage du
5 nombre initial de bits. Le circuit 37 de réglage du nombre initial de bits règle le nombre des données d'entrée initiales pour un code devant être décodé ultérieurement à partir des données de sortie B_2 et B_3 situées dans la bascule bistable de données 17 et ce nombre n'est pas fixé. La
10 mémoire 35 de réglage du nombre fixe de bits est utilisée pour introduire un nombre fixe de données d'entrée au cours du décodage.

On va décrire le fonctionnement de l'appareil représenté sur la figure 10 en rapport avec le cas de l'introduction d'une donnée d'entrée "1101000" représentée sur la figure 9. Pour démarrer l'opération de décodage, un signal de
15 démarrage de décodage est appliqué par l'intermédiaire du conducteur 11 au circuit 12 de détection de fin de décodage de manière à ramener à l'état initial la bascule bistable 14
20 par l'intermédiaire de la ligne de transmission de signaux 32, en plaçant la mémoire à tables de décodage 16 dans l'état de lecture du code blanc. Simultanément la bascule bistable d'adresses de mémoire 15 est également remise à l'état initial de manière à délivrer une donnée (000) depuis
25 les mémoires 16 en direction de la bascule bistable de données 17. D'autre part le circuit 12 de détection de fin de décodage délivre des niveaux bas dans les lignes de sortie 18 et 32 au moyen du signal de démarrage du décodeur de manière à commuter les sélecteurs de données 20 et 21 sur le
30 côté b comme dans le cas de la figure 7. Lors de ce fonctionnement, le sélecteur de données 20 est raccordé au circuit 37 de réglage du nombre initial de bits. Le circuit 37 de réglage du nombre initial de bits est ramené à l'état initial par le signal de démarrage de décodage de manière à positionner
35 une donnée de comptage d'entrée "4" dans le compteur 24 des données d'entrée. Les quatre premiers bits "1101" des données d'entrée sont appliqués au convertisseur série-parallèle des données d'entrée 26 à partir du conducteur 27 en synchronisme avec des impulsions d'horloge d'entrée

provenant de la borne 25. Les données introduites peuvent être représentées par (OD) dans la notation hexadécimale. Ensuite le convertisseur série-parallèle des données d'entrée 26 délivre la donnée d'entrée "1101" au circuit additionneur à huit bits 22. Pendant ce temps, le bit de poids le plus faible $B_1=0$ de la donnée initiale (000) introduit dans la bascule bistable de données 17 est appliqué au circuit 12 de décodage de fin de décodage de manière à fournir un niveau élevé dans la ligne de sortie 18 par l'intermédiaire de la porte OU 33, en provoquant la commutation du sélecteur 21 sur le côté a. Par conséquent les bits B_{10} à B_5 de la donnée initiale (000) sont introduits à partir de la bascule bistable de données 17 dans le circuit additionneur à huit bits 22 par l'intermédiaire du sélecteur de données 21. La donnée d'entrée "1101", c'est-à-dire (OD), et la donnée initiale (00), c'est-à-dire (000) - " $B_4B_3B_2B_1$ " sont ajoutées de manière à fournir l'adresse de saut <OD>, qui est délivrée à la bascule bistable d'adresses de mémoire 15. Le sélecteur de données 20 est commandé par le circuit 39 d'allongement des impulsions en étant commuté sur le côté a après le sélecteur de données 21, c'est-à-dire après la délivrance de l'adresse de saut <OD>. Sous l'effet du signal délivré dans la ligne de transmission de signaux 28 par le circuit additionneur à huit bits 22, une donnée (A00) ($B_{10}B_9B_8B_7B_6B_5XXB_4B_3B_2B_1="101000000000"$ en représentation binaire) correspondant à l'adresse de saut <OD> est lue hors de la mémoire à table de décalage 16 pour être introduite dans la bascule bistable des données 17. Cette donnée est une donnée intermédiaire pour un noeud interne étant donné que le bit de poids le plus faible B_1 est "0". Le circuit 12 de détection de fin de décodage maintient les sélecteurs de données 20 et 21 dans l'état "non fini" (sur le côté a).

Par conséquent "4" est introduit par la mémoire 35 de réglage du nombre fixé de bits dans le compteur des données d'entrée 24 et quatre bits "1000" de la donnée d'entrée suivante sont introduits dans le convertisseur série-parallèle des données d'entrée 26 en synchronisme avec les impulsions d'horloge d'entrée. Le signal de sortie du convertisseur série-parallèle des données d'entrée 26 passe à l'état

"1000" et le signal d'entrée aboutissant au circuit additionneur à huit bits 22 devient "00001000" (08 dans la représentation hexadécimale). Les bits B_5 à B_{10} du signal de sortie provenant de la bascule bistable des données 17 sont délivrés au circuit additionneur à huit bits 22 de manière à envoyer un autre signal d'entrée (A0) à ce dernier. Dans le circuit additionneur à huit bits 22, (08) et (A0) sont ajoutés l'un à l'autre de manière à fournir une adresse de saut <A8>, qui est envoyée à la bascule bistable d'adresses de mémoire 15. Simultanément, la donnée (049) ($B_{10}B_9B_8B_7B_6B_5B_4B_3B_2B_1$ ="000001001001" selon la représentation binaire) est lue hors de la mémoire à tables de décodage 16 pour être introduite dans la bascule bistable des données 17. Le bit de poids le plus faible B_1 ="1" indique que cette donnée est une donnée finale, c'est-à-dire qu'un noeud final a été atteint à la fin de l'opération de décodage. Les sélecteurs de données 20 et 21 sont commutés par le circuit 12 de détection de fin de décodage sur le côté b ("fin"). Etant donné que le bit B_4 est "1", le sélecteur de données 29 est commuté sur le côté MC, c'est-à-dire que le signal de sortie provenant du sélecteur de données 29 est introduit en tant que code de reprise dans la bascule bistable 31 de sortie de la longueur parcourue de manière à obtenir "000001000000" selon la représentation binaire, à savoir la longueur de parcours "64", ce qui termine l'opération de décodage.

Une donnée B_3B_2 ="00" de la bascule bistable des données 17 est envoyée au circuit 37 de réglage du nombre initial de bits, où le "00" est incrémenté d'un un pour fournir une valeur "1", qui représente une différence entre le nombre fixé de bits de la donnée d'entrée "4" et le nombre de bits "3" de la donnée "000" qui n'a pas été utilisée pour le décodage de la donnée d'entrée "11011000". La valeur "1" est appliquée en tant que donnée initiale au compteur des données d'entrée 24. Dans cet exemple, étant donné que le bit B_4 de la donnée finale lue est "1", le circuit 12 de détection de fin de décodage ramène à l'état initial la bascule bistable d'adresses de mémoire 15 tout en maintenant la bascule bistable 14 dans un état inchangé, c'est-à-dire en

maintenant l'état de décodage du code blanc. A partir de la mémoire 16 se trouve lue la donnée (000) qui est introduite dans la bascule bistable des données 17. Ensuite, lors de l'entrée des données dans le convertisseur série-parallèle des données d'entrée 26, le décodage de la donnée d'entrée suivante est effectué de la même manière que dans le cas de l'entrée "11011". Dans le cas où le bit B_4 est "0" lorsque la donnée finale est lue, le circuit 12 de détection de fin de décodage inverse la bascule bistable 14 de manière à commuter la lecture de la mémoire à tables de décodage 16 sur la lecture de la table du code noir et ramène à l'état initial la bascule bistable d'adresses de mémoire 15. Dans la forme de réalisation décrite ci-dessus en liaison avec les figures 8 et 10, la donnée est introduite par échelons d'un nombre fixé de bits, quatre bits dans le cas de cet exemple, au cours du décodage, et ces bits de la donnée à quatre bits introduite, qui n'ont pas été utilisés pour le décodage, c'est-à-dire les trois bits "000" dans l'exemple ci-dessus, sont utilisés pour le décodage immédiatement suivant. Par conséquent le décodage reprend en utilisant ces trois bits et un bit de la donnée suivante (le nombre fixé de bits, c'est-à-dire quatre bits dans cet exemple).

L'appareil de décodage conforme à la présente invention peut être constitué en utilisant non seulement un circuit logique câblé comme représenté sur les figures 7 et 10, mais également un circuit à utilisation générale, muni d'une fonction d'accès de mémoire et d'une fonction de décision et de calcul des données, telle que par exemple un microprocesseur. En se référant à la figure 11, on va donner une description d'un exemple de l'appareil de décodage utilisant un microprocesseur en liaison avec le cas de la table de décodage de la figure 4 (4A, 4B) utilisé. A un bus 41 se trouvent raccordés un microprocesseur 42, un convertisseur série-parallèle 43, une mémoire-tampon de réception 44, une mémoire de programmes 45, un accès de sortie 46 et la mémoire à tables de décodage 16. Dans cette mémoire 16 se trouve mémorisée la table de décodage de la figure 4 (4A, 4B), c'est-à-dire que la donnée représentée par (XX) est mémorisée au niveau de chaque adresse représentée par <XX>. Le

microprocesseur 42 possède plusieurs registres généraux 47, accède à la mémoire de programmes 45 selon un ordre séquentiel de manière à introduire les instructions lues à l'intérieur d'un registre d'instructions 48, décode ces instructions dans un décodeur 49 et exécute différentes opérations et décisions de commande d'entrée/sortie basées sur les résultats décodés. Les opérations logiques d'addition et de décision sont effectuées par un opérateur 51 en liaison avec les données du registre général 47. Les résultats des opérations sont mémorisés dans le registre général 47. La mémoire à tables de décodage 16 et la mémoire de programmes 46 peuvent être constituées par une mémoire. Un train de codes d'entrée est introduit dans le convertisseur série-parallèle 43 à partir du conducteur 27. Lors de chaque application de par exemple huit bits, un signal d'interruption est appliqué par l'intermédiaire d'une ligne de transmission d'interruptions 52 à un circuit d'interruption 53 du microprocesseur 42. Lorsqu'il reçoit le signal d'interruption, le microprocesseur 42 mémorise une donnée série à huit bits en provenance du convertisseur série-parallèle 43 dans la mémoire-tampon de réception 44. L'accès de sortie 46 sert à délivrer les résultats décodés.

En se référant maintenant à un organigramme tel que celui représenté sur la figure 12, on va décrire un exemple du fonctionnement de l'appareil de décodage mentionné ci-dessus. Le fonctionnement commence avec un pas de traitement S_1 , au cours duquel le nombre des bits initial d'entrée, c'est-à-dire le nombre des bits devant être introduits en premier à partir du train de codes d'entrée, "4" dans cet exemple, est positionné dans un registre général 47a et une donnée d'adresse initiale pour la mémoire à tables de décodage 16, c'est-à-dire (00) dans cet exemple, est positionnée dans un registre général 47b. Au cours du pas de traitement suivant S_2 se trouve vérifié si par exemple si tous les bits du code d'entrée à huit bits introduits dans un registre général 47c à partir de la mémoire-tampon de réception 44 ont été utilisés ou non. Dans le cas où le code d'entrée a déjà été utilisé, le fonctionnement passe à une phase ou pas de traitement S_3 , au cours duquel la donnée suivante à

huit bits est introduite à partir de la mémoire-tampon de réception 44 dans le registre général 47c. Au cours d'un pas de traitement S_4 , la donnée correspondant au nombre des bits d'entrée "4" positionné dans le registre 47a au cours du pas de traitement S_1 est transférée du registre 47c à un registre général 47d. Au cours d'un pas de traitement S_5 , les données ainsi transférées au registre général 47d et la donnée d'adresse initiale (00) positionnée dans le registre 47b au cours du pas de traitement S_1 , sont ajoutées l'une à l'autre pour fournir une adresse. Cette adresse est utilisée pour l'accès à la mémoire à tables de décodage 16 au cours d'un pas de traitement S_6 . Au cours d'un pas de traitement S_7 il est vérifié si le bit de poids le plus faible B_1 de la donnée lue hors de la mémoire à tables de décodage 16, est "1" ou non. Lorsque le bit B_1 n'est pas "1", la donnée lue est une donnée intermédiaire et la donnée d'adresse suivante et le nombre des bits du code d'entrée devant être introduits ensuite sont positionnés. C'est-à-dire que, dans l'exemple décrit précédemment, il est vérifié au cours du pas de traitement S_6 si le bit B_2 de la donnée lue est "1" ou non. Si le bit B_2 n'est pas "1", alors le nombre des bits du code d'entrée devant être introduits ensuite est "1" et, au cours d'un pas de traitement S_9 , les bits B_3 à B_8 des données lues hors de la mémoire à tables de décodage 16 sont positionnées sous la forme d'une donnée d'adresse dans le registre général 47b et, simultanément, le nombre des bits du code d'entrée "1" est positionné dans le registre général 47a. Dans le cas où il est décidé, au cours d'un pas de traitement S_8 , que le bit B_2 est "1", le nombre des bits du code d'entrée devant être introduits ensuite est "2", et, dans un pas de traitement S_{10} , les bits B_3 à B_8 de la donnée lue sont chargés dans le registre général 47b et simultanément le nombre des bits suivants du code d'entrée "2" est chargé dans le registre général 47a. Après les pas de traitement S_9 ou S_{10} , le fonctionnement revient au pas de traitement S_2 de manière à répéter la procédure indiquée ci-dessus.

Lorsqu'au pas de traitement S_7 il est décidé que le bit de poids le plus faible B_1 est "1", la donnée lue hors

de la mémoire à tables de décodage 16 est une donnée finale et l'opération passe au pas de traitement S_{11} , au cours duquel se trouve vérifié si le bit B_2 de la donnée est "1" ou non. S'il est décidé que le bit B_2 est "1", la donnée est

5 un code de reprise et n (en nombre décimal) représenté par les bits B_3 à B_8 de la donnée doit être multiplié par 64 pour fournir la longueur de parcours (pas de traitement S_{12}). Dans le cas où il est décidé au cours d'un pas de traitement S_{11} que le bit de poids le plus faible B_1 est "0" et que la

10 donnée lue est un code final, les bits B_3 à B_8 des données sont obtenus sous la forme de la longueur de parcours, lors d'un pas de traitement S_{13} . Le résultat décodé obtenu lors du pas de traitement S_{12} ou S_{13} est délivré en tant que sortie décodée par l'orifice de sortie 46 (figure 11) au cours

15 d'un pas de traitement S_{14} . Ensuite il est vérifié au cours d'un pas de traitement S_{15} si le décodage a été entièrement achevé ou non. Ceci peut être décidé sur la base de la donnée du code d'entrée ou de la donnée décodée, comme dans le cas de l'art antérieur. Lorsqu'il est décidé que le déco-

20 dage n'est pas encore achevé, le fonctionnement passe au pas de traitement S_{16} , lors duquel, lorsqu'il est décidé que le code décodé est un code de terminaison, c'est-à-dire lorsqu'il est décidé au cours du pas de traitement S_{11} que le bit B_2 est "0", l'opération suivante de décodage est commutée

25 entre le décodage du code blanc et le décodage du code noir et réciproquement, puis l'opération revient au pas de traitement S_1 . Dans le cas où il est décidé au cours du pas de traitement S_{15} que le décodage n'est pas encore achevé et que le bit B_2 est "1" au cours du pas de traitement S_{11} , l'opéra-

30 tion revient au pas de traitement S_1 sans passer par le pas de traitement S_{16} .

Un exemple de la table de décodage pour des codes noirs est représenté sur la figure 13 (13A, 13B) de la même manière que sur la figure 4 (4A, 4B). Comme on le comprendra

35 à partir de cette table, il existe un noeud final au niveau d'une seconde branche à partir de la racine 1, de sorte que le nombre des bits du code d'entrée devant être introduits en premier est "2". Comme cela est visible d'après les figures 4 (4A, 4B) et 13 (13A, 13B), ces tables de décodage

contiennent des adresses inutilisées (par exemple <OA> et <OB> sur la figure 13 (13A, 13B)) et par conséquent la mémoire à tables de décodage n'est pas utilisée de façon efficace à ce point de vue. On réalise la mémoire à tables
5 de décodage de telle manière que de telles adresses inutiles de puissent pas exister, il est possible de réduire en supplément la capacité de la mémoire.

Etant donné que la forme de réalisation de la figure 7 utilise une mémoire à un mot à huit bits comme mémoire
10 à tables de décalage 16 et huit bits comme information du résultat décodé et en tant que donnée pour la détermination de l'adresse à laquelle l'accès doit être effectué ensuite, de sorte que le nombre des bits d'entrée devant être introduits tout d'abord pour chaque code, "4", est positionné
15 dans la mémoire 23. Cependant il est possible d'adopter un agencement tel que dans le cas où le nombre des bits devant être introduits à la fois, y compris le nombre des bits devant être introduits tout d'abord, est "2" ou moins, le nombre des bits devant être introduits est toujours déterminé
20 par l'information qui détermine le nombre des bits d'entrée dans la donnée lue à partir de la mémoire à tables de décodage 16. En choisissant pour le nombre des bits constituant un mot de la mémoire à tables de décodage 16 une valeur supérieure à 8, il est possible que dans le cas où
25 le nombre des bits devant être introduits à la fois, y compris le nombre des bits devant être introduits en premier, est supérieur à deux, le nombre des bits devant être introduits conformément à un nombre maximum de bits d'entrée soit encore toujours déterminé par l'information qui détermine le
30 nombre des bits d'entrée dans la donnée lue à partir de la mémoire à tables de décodage 16.

Grâce à l'appareil de décodage selon la présente invention, décrit ci-dessus, il est possible de réduire la capacité requise de la mémoire à tables de décodage. La figure 14 montre des capacités de mémoire requises pour différents procédés de décodage. Les abscisses représentent les
35 procédés de décodage. L'abscisse 1 indique un procédé de décodage utilisant une table de codes et selon lequel les codes de la table de codes mémorisés sont séquentiellement

produits en vue d'être comparés à un code d'entrée pour obtenir un résultat décodé basé sur la coïncidence entre les codes. Les abscisses 2, 3 et 4 représentent chacune le cas de l'utilisation du procédé décrit précédemment en liaison
5 avec la figure 7, l'abscisse 2 représentant le cas où le nombre des bits du code d'entrée est choisi à une valeur préférée pour chaque noeud de l'arbre de codes, l'abscisse 3 illustrant le cas où le nombre des bits devant être introduits à la fois est choisi étant égal soit à "1", soit à "2",
10 soit à "4", et l'abscisse 4 représentant le cas où le nombre de bits devant être introduits à la fois est choisi comme étant égal soit à "1", soit à "2". L'abscisse 5 repère le procédé de décodage bit par bit décrit précédemment en référence à la figure 1. Les abscisses 6 à 10 illustrent chacune
15 le cas de l'utilisation du procédé de décodage décrit précédemment en référence à la figure 10. Pour les abscisses 6 à 10, le nombre fixé des bits devant être introduits à la fois est choisi comme étant égal à "2", "3", "4", "5", "6" respectivement. Les abscisses 11 à 17 illustrent chacune le cas de
20 l'utilisation du procédé de décodage considéré en référence précédemment en liaison avec la figure 3. Pour ces cas 11 à 17, le nombre des bits devant être introduits à la fois est choisi comme étant égal à "7", "8", "9", "10", "11", "12" et "13" respectivement. L'ordonnée représente la capacité
25 de mémorisation. La courbe 63 repère le cas de l'utilisation d'une mémoire à 1 mot à 16 bits et la courbe 64 repère le cas de l'utilisation d'une mémoire à un mot à huit bits. Il ressort de la figure 14 que la capacité de mémoire requise pour l'appareil de la figure 7 est égale à la plus petite
30 des capacités de mémoire requises dans les procédés de l'art antérieur, et est égale en particulier à environ 1/3 de la plus petite des capacités dans les cas de l'utilisation du procédé de la figure 3. Dans l'appareil représenté sur la figure 10, un accroissement du nombre des bits devant
35 être introduits en premier provoque un accroissement de la capacité de mémoire requise, mais la capacité de mémoire est inférieure ou sensiblement égale à la capacité de mémoire nécessaire dans le cas de la figure 3.

La figure 15 indique les temps d'accès à la

mémoire, qui sont requis pour les différents procédés de décodage. Les abscisses repèrent les mêmes procédés que ceux mentionnés ci-dessus en rapport avec la figure 14 et les ordonnées représentent le nombre d'accès. Les courbes 65 et 65' indiquent chacune un temps d'accès maximum, les courbes 66 et 66' un temps d'accès moyen et les courbes 67 et 67' un temps d'accès moyen pondéré prenant en compte la probabilité d'apparition des différents codes ; les courbes 65', 66' et 67' illustrent le cas de l'utilisation d'une mémoire à un mot à 16 bits et les autres courbes illustrent le cas de l'utilisation d'une mémoire à un mot à huit bits. Sur la figure 15 on note que le nombre des actions nécessaires dans l'appareil conforme à la présente invention est nettement inférieur au nombre des actions nécessaires dans le cas du procédé de décodage bit par bit repéré à l'abscisse 5, et est seulement légèrement supérieur à celui du cas de l'appareil classique à la figure 3. En considérant les temps d'accès moyens pondérés qui représentent les valeurs les plus usuelles, l'appareil selon la présente invention est assez proche de l'appareil de la figure 3.

Sur la base des données indiquées ci-dessus, la figure 16 montre la capacité de la mémoire à tables de décodage et les temps d'accès dans chacun des différents appareils, la capacité de la mémoire à tables de décodage et le nombre d'accès dans l'appareil de la figure 7 étant représentés respectivement par 1.

Comme cela a été décrit précédemment, conformément à la présente invention, le nombre des bits devant être introduits ensuite est déterminé par lecture hors de la mémoire à tables de décodage, par une addition ou une opération analogue de la donnée lue hors de la mémoire à tables de décodage et le code introduit est mis en oeuvre pour déterminer l'adresse à laquelle l'accès doit être ensuite effectué, puis l'accès à la même mémoire de décodage est effectué en utilisant l'adresse. Ceci réduit la capacité de la mémoire et le nombre des accès et par conséquent permet un décodage à grande vitesse avec un matériel de taille réduite. En outre, conformément à la forme de réalisation de la figure 7, dans le cas du décodage d'un code, le bit de

tête du code d'entrée devient le premier bit du code suivant, ce qui facilite le traitement ultérieur.

Bien que, dans ce qui précède, l'adresse, à laquelle on accède ensuite, est obtenue par addition l'un à l'autre
5 de la donnée lue hors de la mémoire à tables de décodage et du code d'entrée, il est également possible d'obtenir cette adresse par d'autres opérations appliquées à ces données, comme par exemple une soustraction et une multiplication. Dans ce qui précède, étant donné que les formes de réalisation
10 tion de la présente invention ont été décrites comme étant appliquées au décodage du code de Huffman modifié, l'information représentant le type du code, c'est-à-dire le fait que le code est un code de reprise ou un code de terminaison ou de fin, est inclus dans le code final, mais dans le cas d'un
15 code de Huffman simple, l'information indiquant le type de code n'est pas nécessaire. En outre la présente invention peut être également appliquée au décodage de codes d'une longueur variable représentée d'une manière générale par un arbre de code .

20 Il ressort à l'évidence que l'on peut apporter de nombreuses modifications et variantes à l'objet de l'invention sans sortir du cadre de cette dernière.

REVENDECATIONS

1. Appareil permettant le décodage d'un code de longueur variable représenté par un arbre de code , dans lequel un dispositif de lecture à accès à une mémoire à tables
5 de décodage (16) pour en extraire des données par lecture, des moyens de décision décident si les données extraites par lecture sont soit des données finales, soit des données intermédiaires, et, dans le cas de données finales, une information indiquant le résultat du décodage dans
10 les données finales et délivrée par des moyens de sortie (31), et, dans le cas de données intermédiaires, la mémoire à tables de décodage (16) est lue sur la base de la donnée d'adresse présente dans les données intermédiaires, caractérisé en ce que la mémoire à tables de décodage (16) possède
15 une zone destinée à la mémorisation des données finales incluant l'information indiquant l'achèvement du décodage et une information indiquant le résultat décodé, et une zone destinée à la mémorisation des données intermédiaires incluant une information indiquant un décodage inachevé, une
20 information indiquant le nombre des bits devant être introduits à partir d'un train de codes d'entrée et une information destinée à déterminer l'adresse de la mémoire à tables de décodage, à laquelle doit s'effectuer l'accès, et qu'il est prévu des moyens d'entrée (24, 26, 36) permettant d'introduire, dans le cas des données intermédiaires, une donnée
25 provenant du train de codes d'entrée au moyen du nombre de bits indiqué par l'information des données intermédiaires indiquant le nombre des bits devant être introduits et des moyens (22) de traitement des adresses destinés à traiter, dans le cas des données intermédiaires, l'information des
30 données intermédiaires pour la détermination de l'adresse de la mémoire à tables de décodage, à laquelle doit être effectué l'accès, et de la donnée introduite par les moyens d'entrée pour obtenir l'adresse de la mémoire à tables de
35 décodage, à laquelle l'accès doit se faire ensuite.

2. Appareil de décodage selon la revendication 1, caractérisé en ce que l'information indiquant le nombre de bits devant être introduits est choisie comme étant un nombre variable inférieur au nombre des branches partant d'une

racine ou d'un noeud interne et aboutissant au noeud final le plus proche de ce dernier dans l'arbre de code .

3. Appareil de décodage selon la revendication 1, caractérisé en ce que le nombre des bits devant être introduits en premier pour chaque code du train de codes d'entrée est mémorisé dans une mémoire de données initiales prévue indépendamment de la mémoire à tables de décodage (16).

4. Appareil permettant le décodage d'un code de longueur variable représenté par un arbre de code , dans lequel un dispositif de lecture a accès à une mémoire à tables de décodage (16) pour en extraire des données par lecture, des moyens de décision décident si les données extraites par lecture sont soit des données finales, soit des données intermédiaires, et, dans le cas de données finales, une information indiquant le résultat du décodage dans les données finales est délivrée par des moyens de sortie (31), et, dans le cas de données intermédiaires, la mémoire à tables de décodage (16) est lue sur la base de la donnée d'adresse présente dans les données intermédiaires, caractérisé en ce que la mémoire à tables de décodage (16) comporte une zone destinée à la mémorisation des données finales incluant une information indiquant l'achèvement du décodage, une information indiquant un résultat décodé et une information indiquant le nombre des bits devant être introduits en premier pour chaque code, et une zone destinée à la mémorisation des données intermédiaires incluant une information indiquant un décodage inachevé et une information pour la détermination de l'adresse de la mémoire à tables de décodage, à laquelle l'accès doit s'effectuer, et qu'il est prévu des moyens d'entrée (24, 26, 35) pour introduire une donnée d'un nombre fixé de bits m (m étant un nombre entier supérieur ou égal à 2) à partir du train de codes d'entrée dans le cas des données intermédiaires, des moyens (22) de traitement des adresses pour traiter, dans le cas des données intermédiaires, l'information pour déterminer l'adresse de la mémoire à tables de décodage à laquelle l'accès doit être effectué, et la donnée introduite par les moyens d'entrée pour obtenir l'adresse de la mémoire à tables de décodage, à laquelle l'accès doit se faire ensuite, des moyens

(12) pour activer de façon répétée les moyens de lecture, les moyens d'entrée, les moyens de décision et les moyens de traitement des adresses jusqu'à ce que la donnée finale soit extraite par lecture, et des seconds moyens d'entrée
5 (24, 26, 37) pour introduire, dans le cas des données finales, une donnée du train de codes d'entrée au moyen du nombre de bits indiqués par l'information indiquant le nombre des bits devant être introduits en premier.

5. Appareil de décodage selon la revendication 4,
10 caractérisé en ce que l'information indiquant le nombre des bits devant être introduits en premier est $L-(n-1)m$, où L est la longueur du code décodée à cet instant et m est le nombre des accès effectués à la mémoire à tables de décodage (16) pour le décodage.

15 6. Appareil de décodage selon la revendication 4, caractérisé en ce qu'il comporte en outre des moyens pour mémoriser un nombre fixe m , à savoir le nombre des bits qui y sont introduits dans le cas où les moyens de décision décident que la donnée extraite par lecture est une donnée
20 intermédiaire.

7. Appareil de décodage selon l'une des revendications 1 ou 4, caractérisé en ce que les données finales comportent une information indiquant le type d'un code décodé.

25 8. Appareil de décodage selon la revendication 7, caractérisé en ce que les moyens de sortie (31) sont agencés de telle manière que dans le cas où les moyens de décision décident que la donnée extraite par lecture est une donnée finale, la destination de l'information du résultat
30 décodé de la donnée finale est modifiée conformément à l'information de la donnée finale indiquant le type de code.

9. Appareil de décodage selon l'une des revendications 1 ou 4, caractérisé en ce que les moyens (22) de traitement des adresses sont des moyens permettant de réaliser
35 une opération d'addition.

10. Appareil de décodage selon l'une des revendications 1 ou 4, caractérisé en ce que, dans les formats des données finales et des données intermédiaires, le même bit est affecté à l'information indiquant l'achèvement du

décodage, et l'information indiquant l'état inachevé du décodage.

11. Appareil de décodage selon l'une des revendications 1 ou 4, caractérisé en ce que, dans les formats des
5 données finales et des données intermédiaires, l'affectation de bits à l'information du résultat décodé et l'affectation de bits à l'information pour la détermination de l'adresse de la mémoire à tables de décodage (16) à laquelle l'accès doit être effectué, se recouvrent au moins récipro-
10 quement.

1/17

FIG. 1

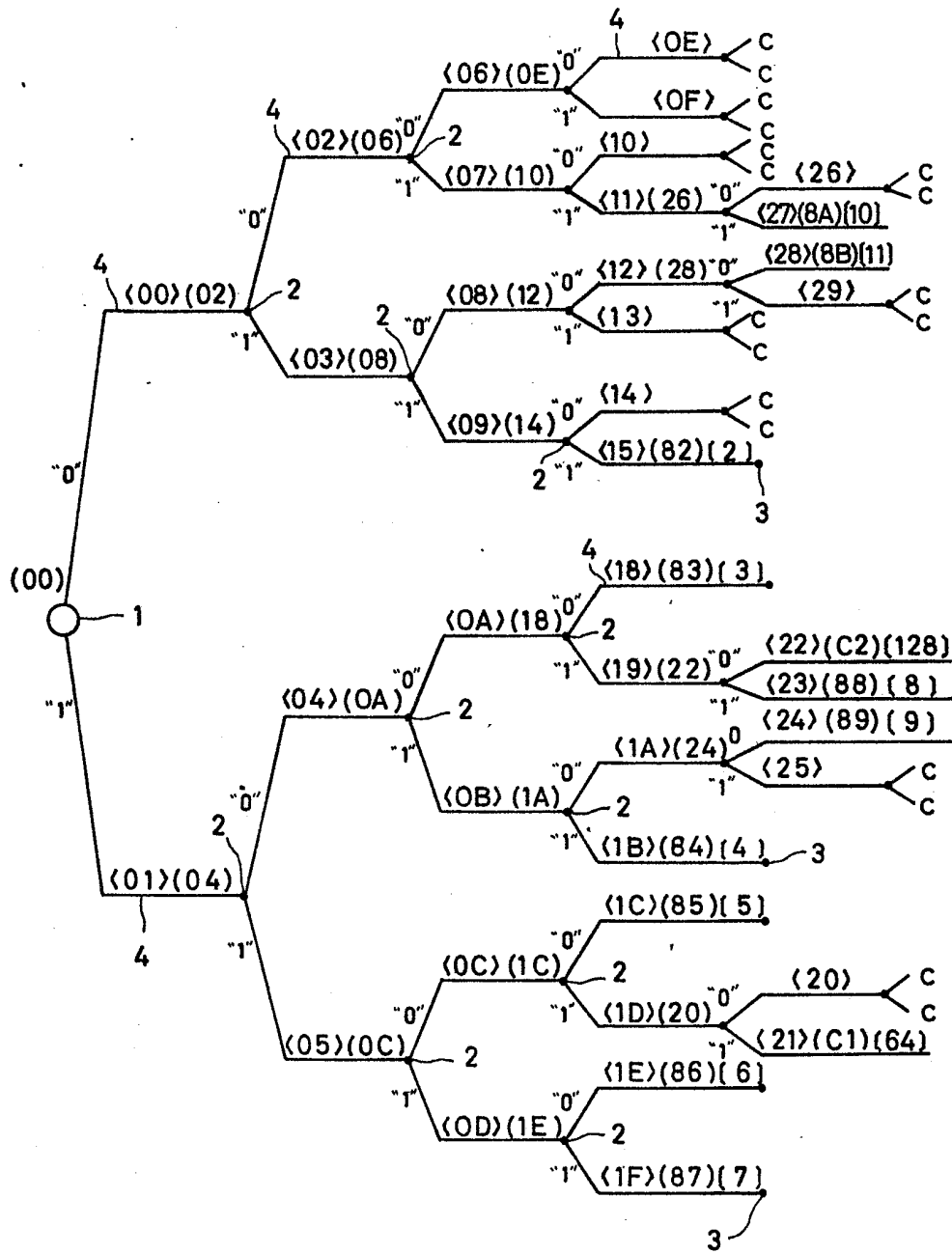
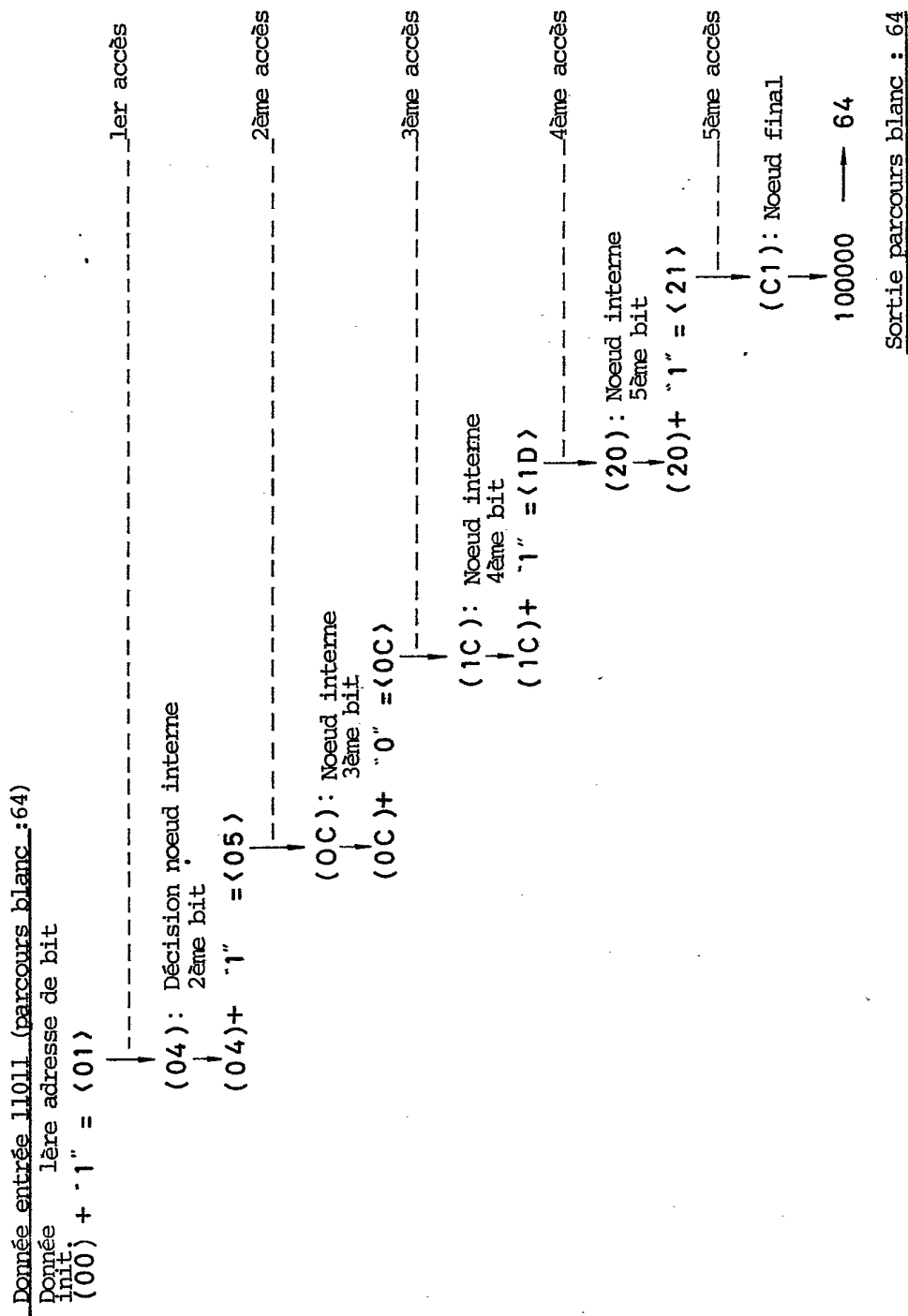
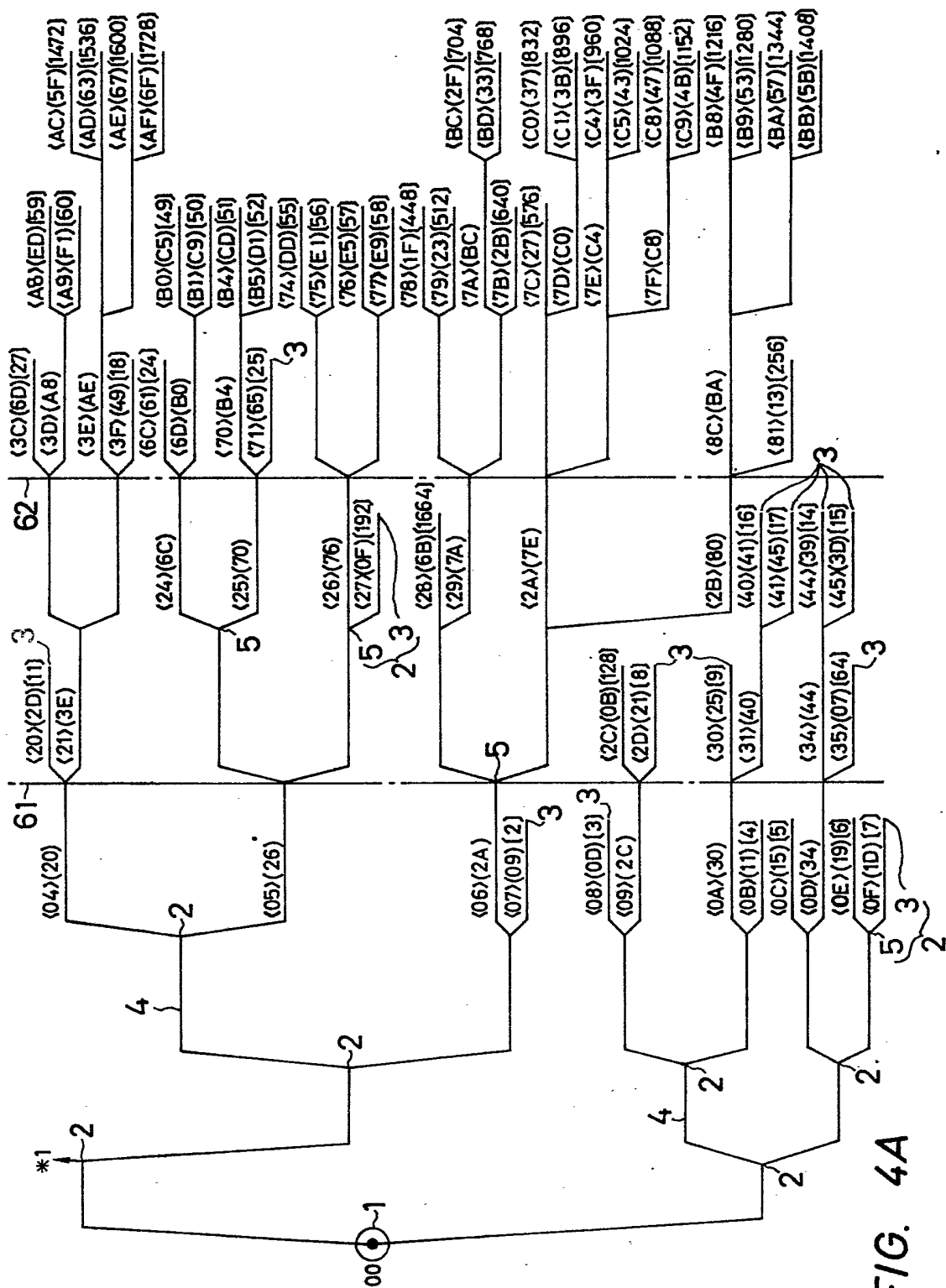


FIG. 2





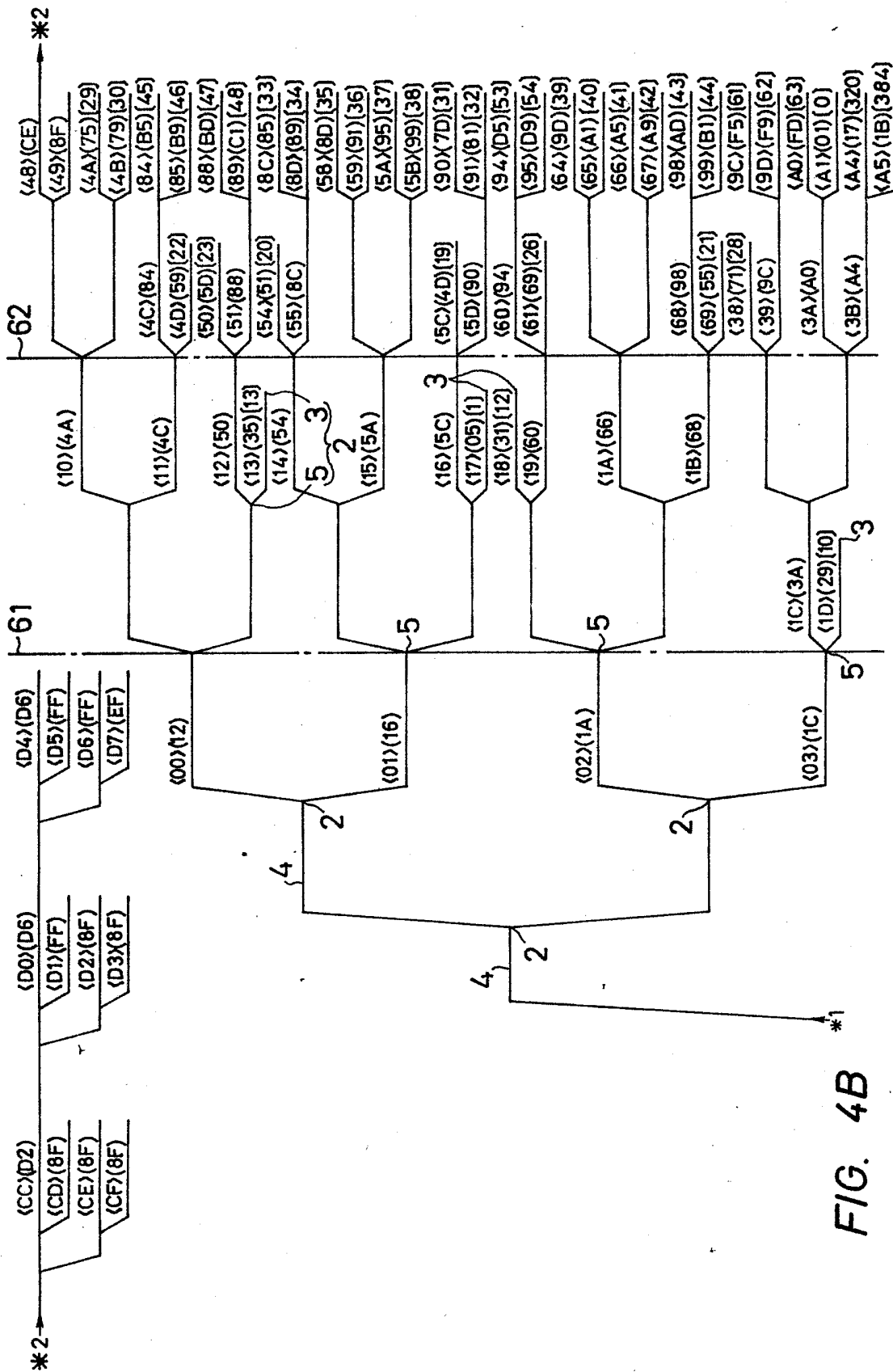


FIG. 4B

FIG. 5A

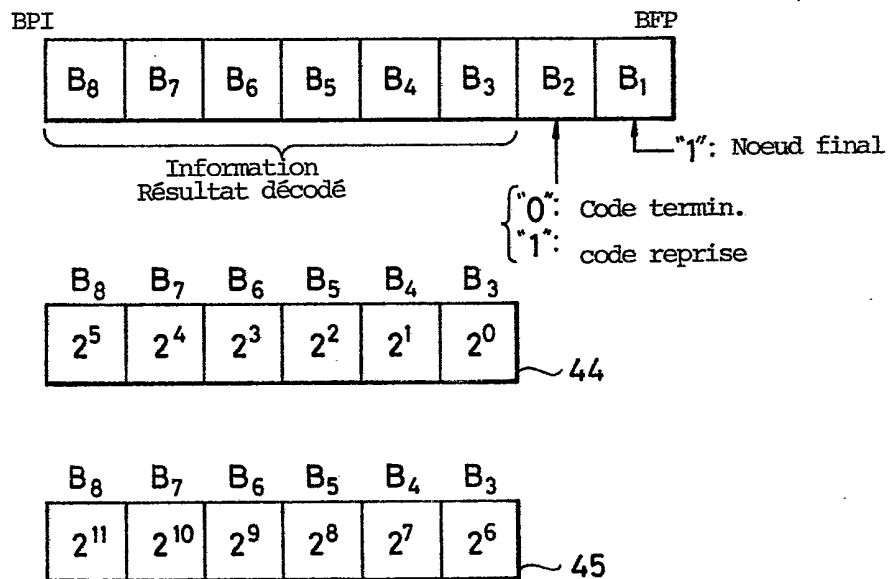


FIG. 5B

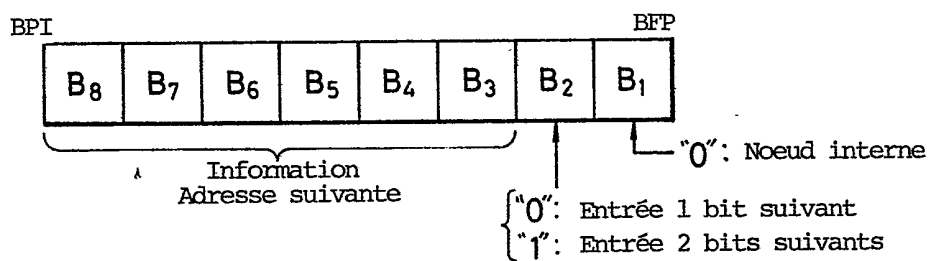


FIG. 6

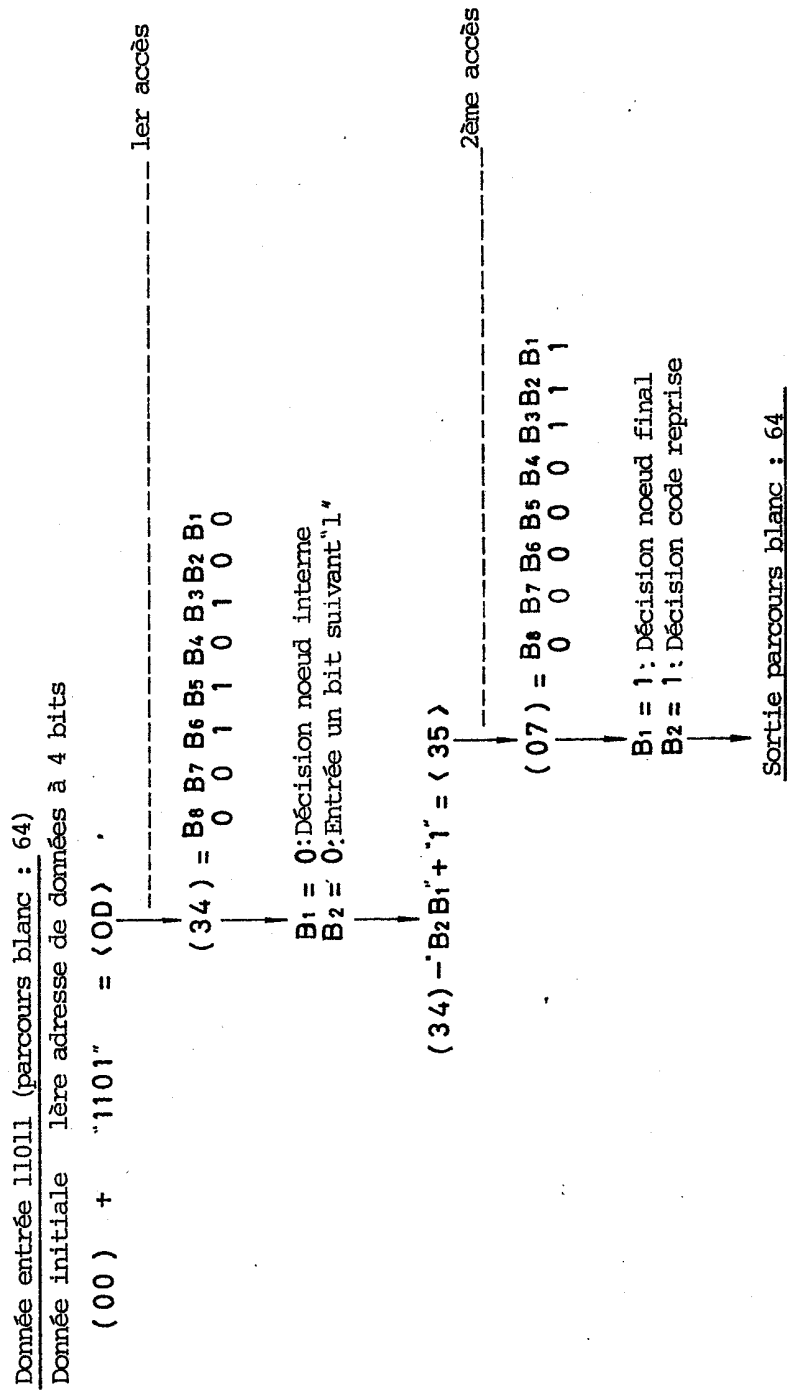
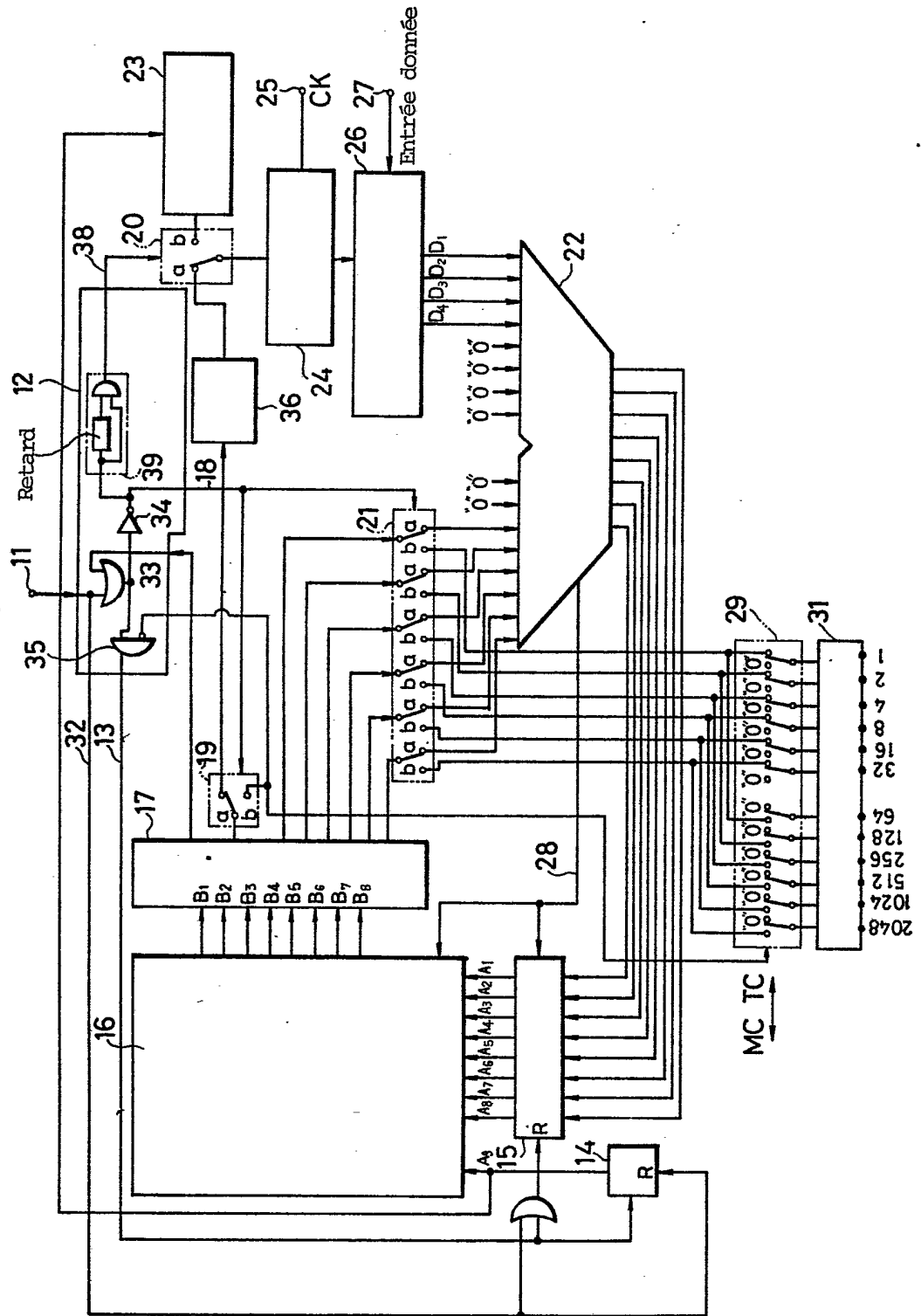
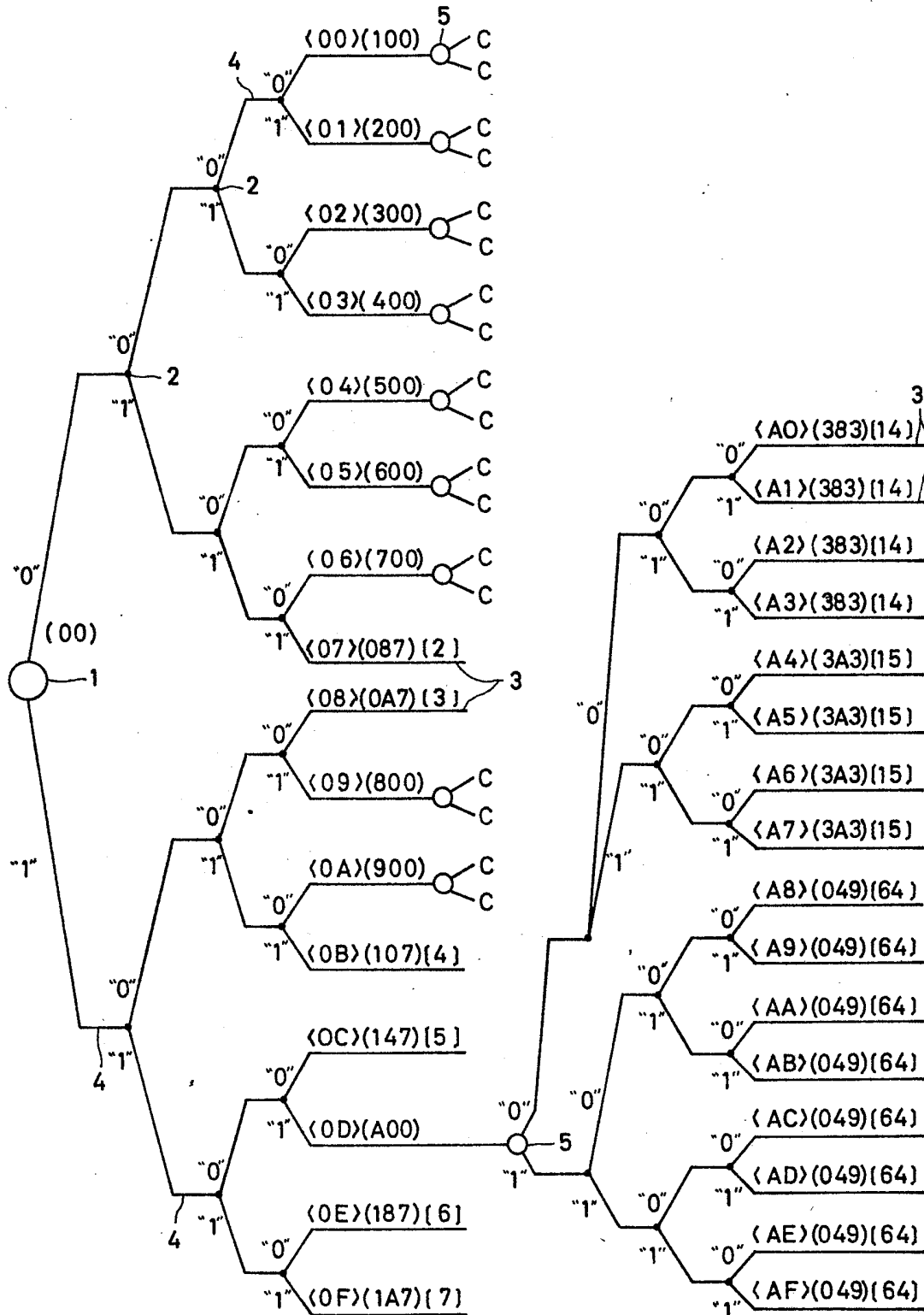


FIG. 7



9/17

FIG. 8



10/17

FIG. 9

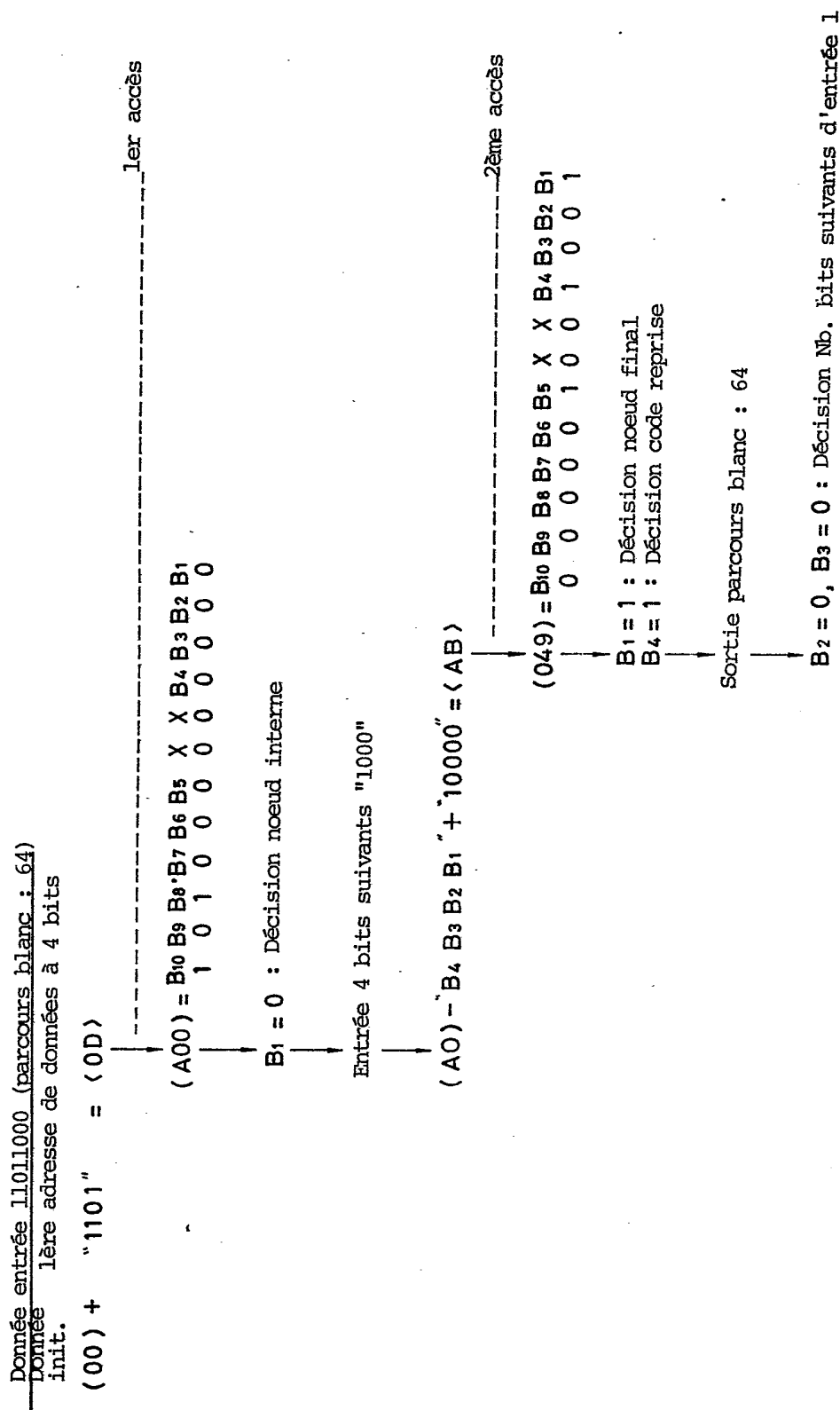


FIG. 10

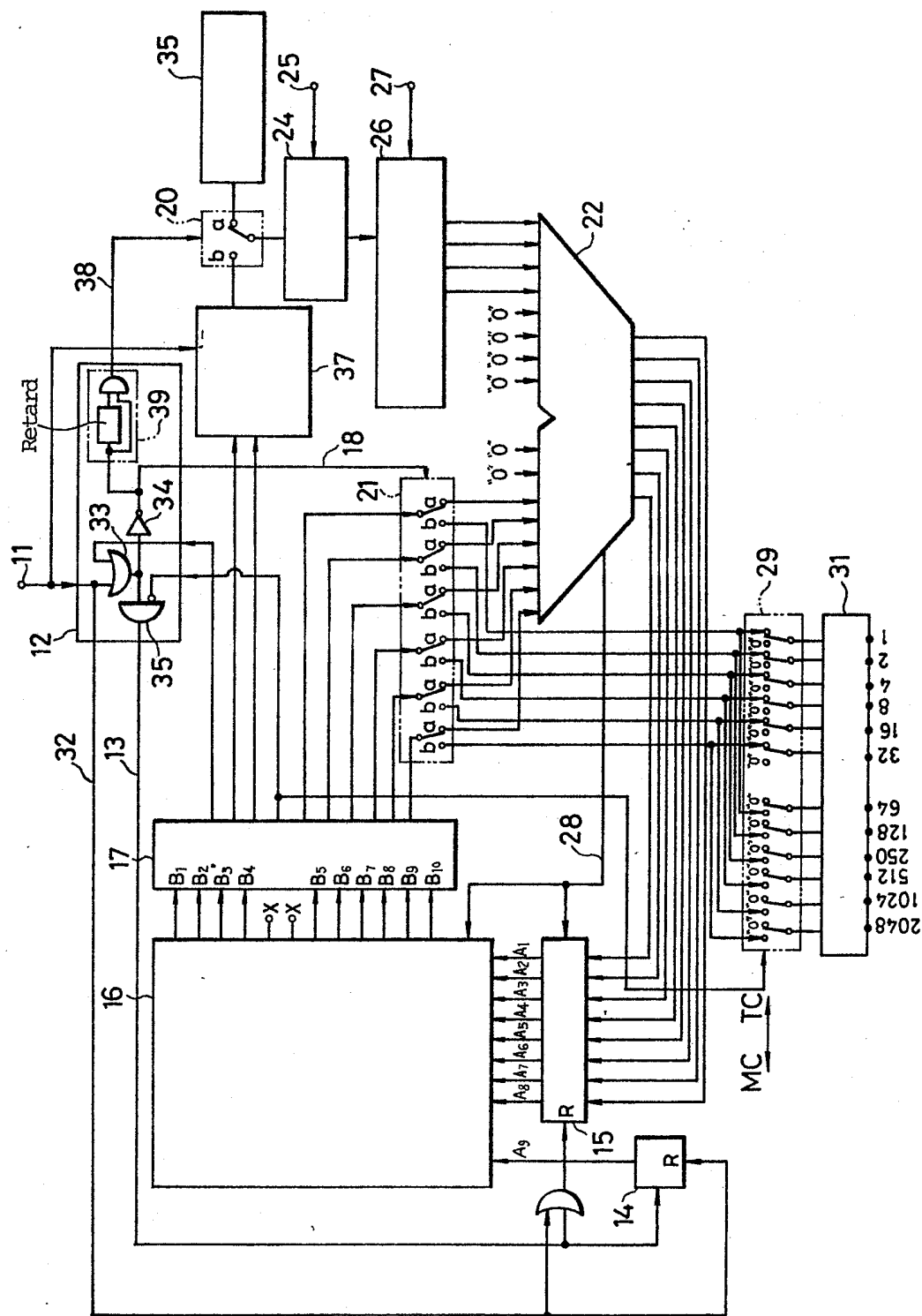


FIG. 11

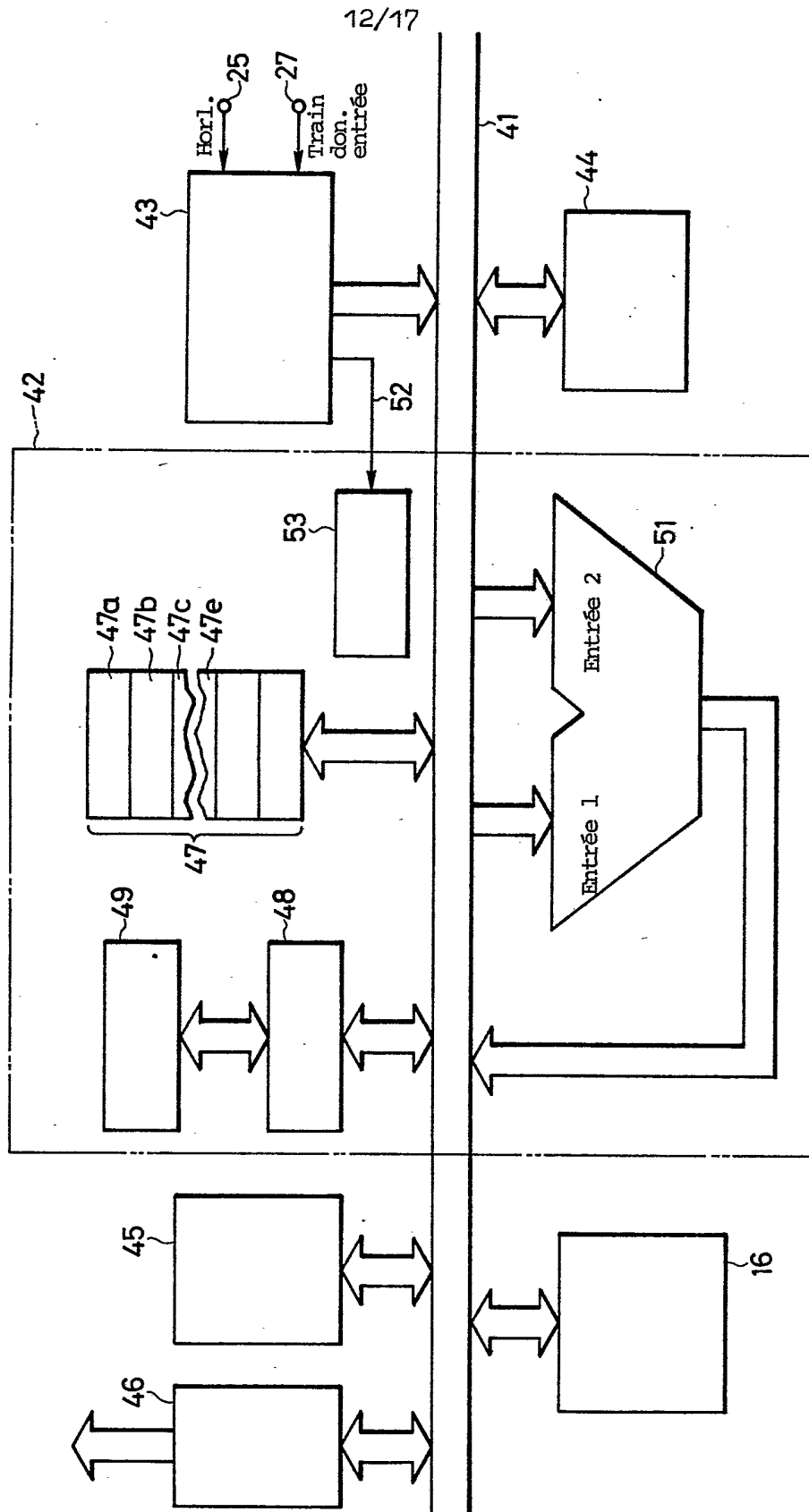
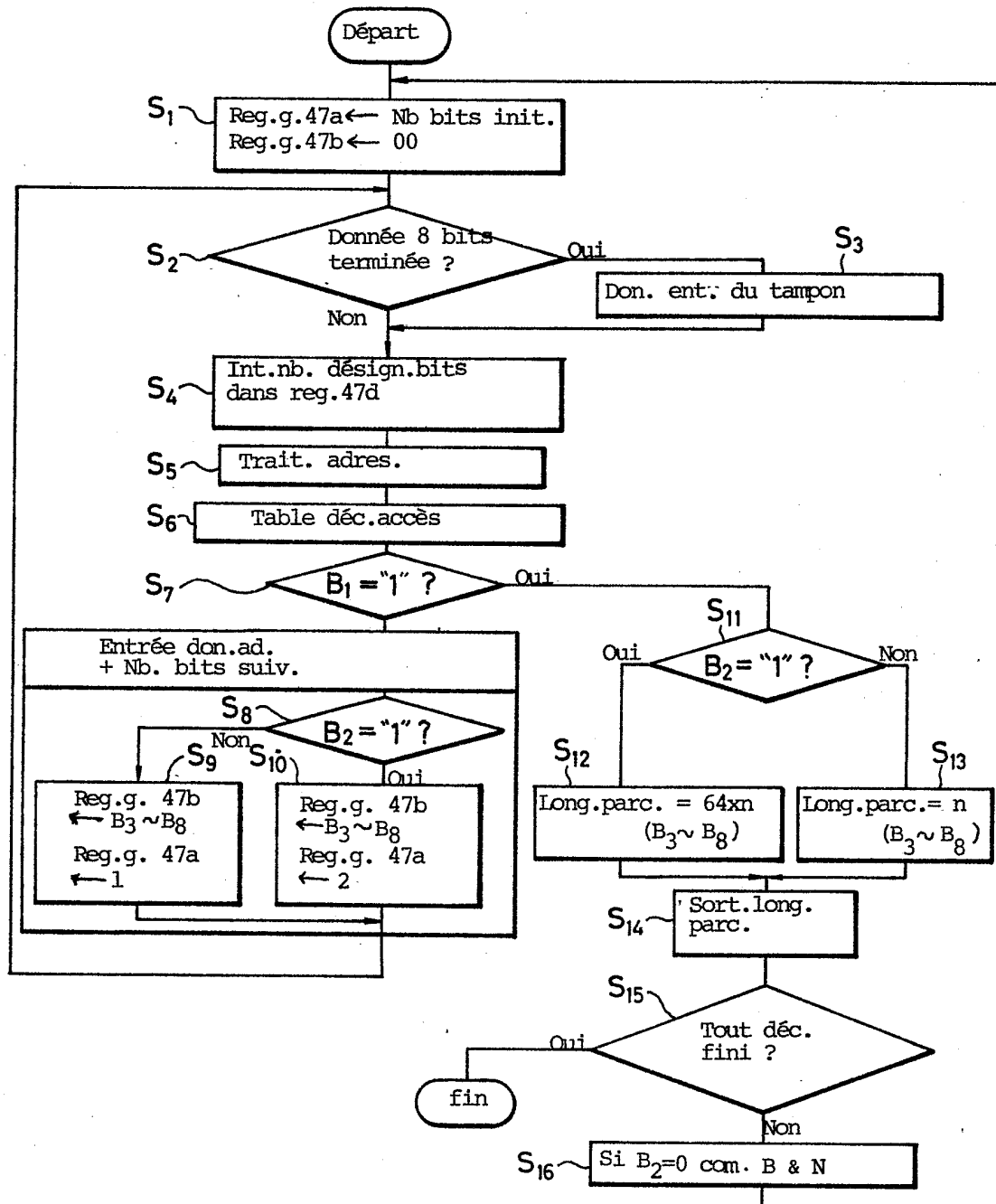
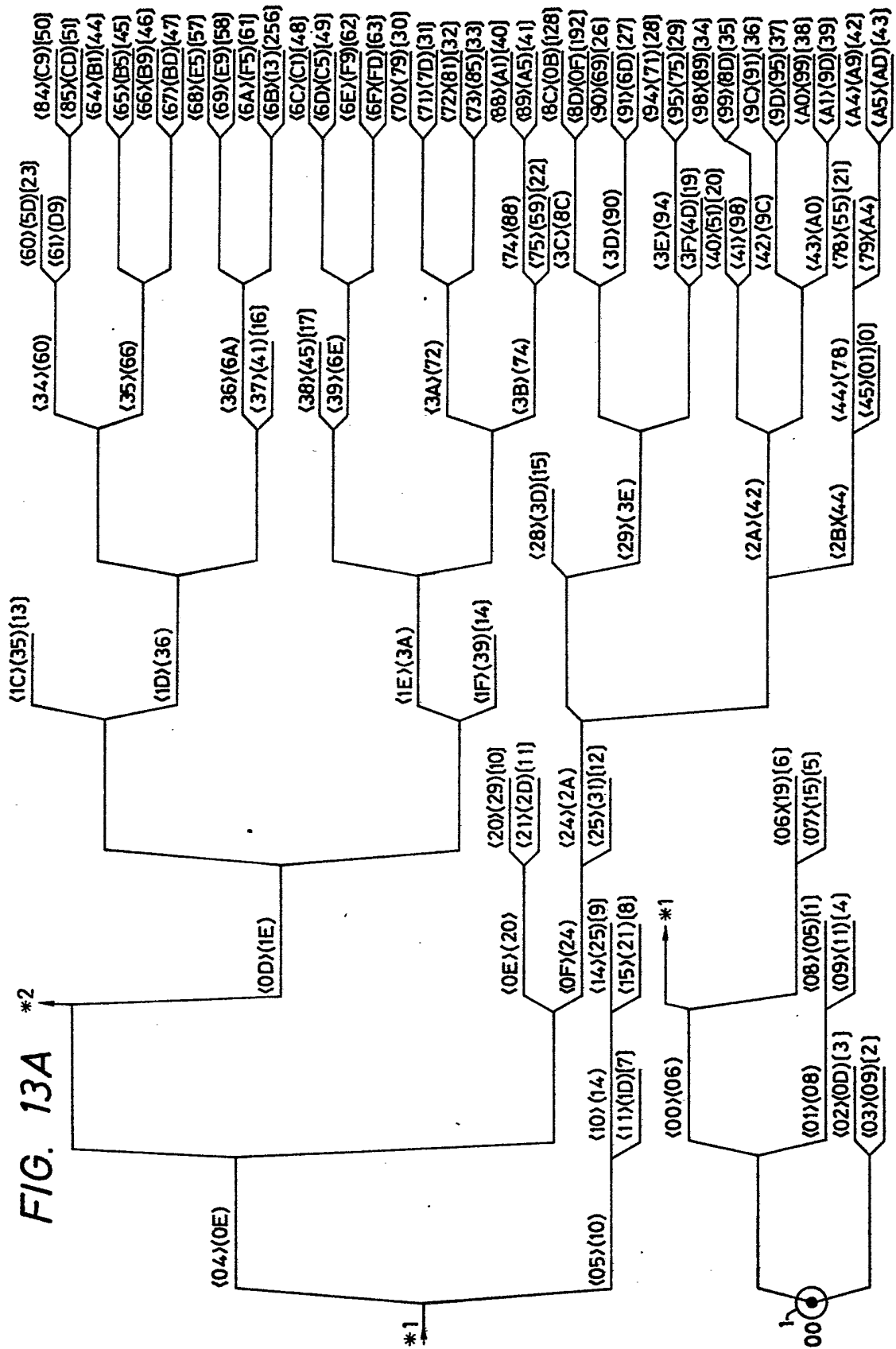


FIG. 12





10. 13B

*EOL

(18)(D2)
(19)(8F)

(D0)(D6)
(D1)(8F)
(D2)(8F)
(D3)(8F)

(D4)(DA)
(D5)(FF)
(D6)(8F)
(D7)(8F)
(D8)(DA)
(D9)(FF)
(DA)(EF)
(DB)(EF)

(1A)(2E)

(2C)(49)(18)
(2D)(4A)

(48)(D1)(52)
(49)(A8)
(4A)(AC)
(4B)(DD)(55)
(4C)(E1)(56)
(4D)(B0)
(4E)(B4)
(4F)(ED)(59)
(7C)(F1)(60)
(7D)(B8)
(80)(BC)
(81)(17)(320)
(58)(1B)(384)
(59)(1F)(448)
(5A)(C0)
(5B)(D5)(53)
(5C)(D9)(54)
(5D)(C4)
(5E)(C8)
(5F)(CC)

(2E)(4E)
(2F)(50)
(30)(54)
(31)(5A)
(32)(5E)
(33)(07)(64)

(50)(7C)
(51)(61)(24)
(54)(65)(25)
(55)(80)

(A8)(2B)(640)
(A9)(2F)(704)
(AC)(33)(768)
(AD)(37)(832)
(BO)(53)(1280)
(B1)(57)(1344)
(B4)(5B)(1408)
(B5)(5F)(1472)
(B8)(63)(1536)
(B9)(69)(1600)
(BC)(6B)(1664)
(BD)(6F)(1728)

(1B)(32)

(C0)(23)(512)
(C1)(27)(576)
(C4)(3B)(896)
(C5)(3F)(960)
(C8)(43)(1024)
(C9)(47)(1088)
(CC)(4B)(1152)
(CD)(4F)(1216)

(0C)(1A)

*2

FIG. 14

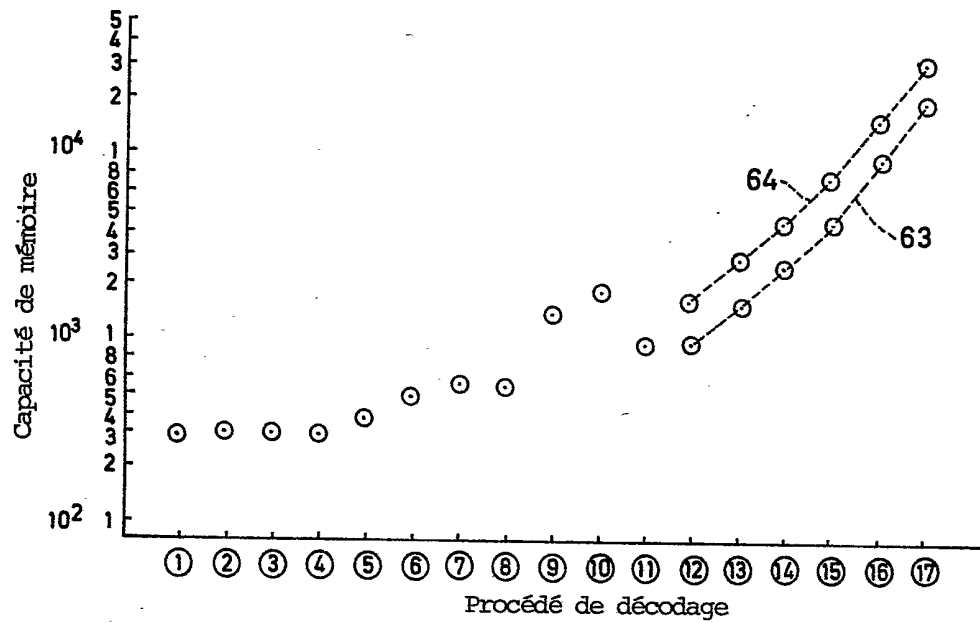


FIG. 15

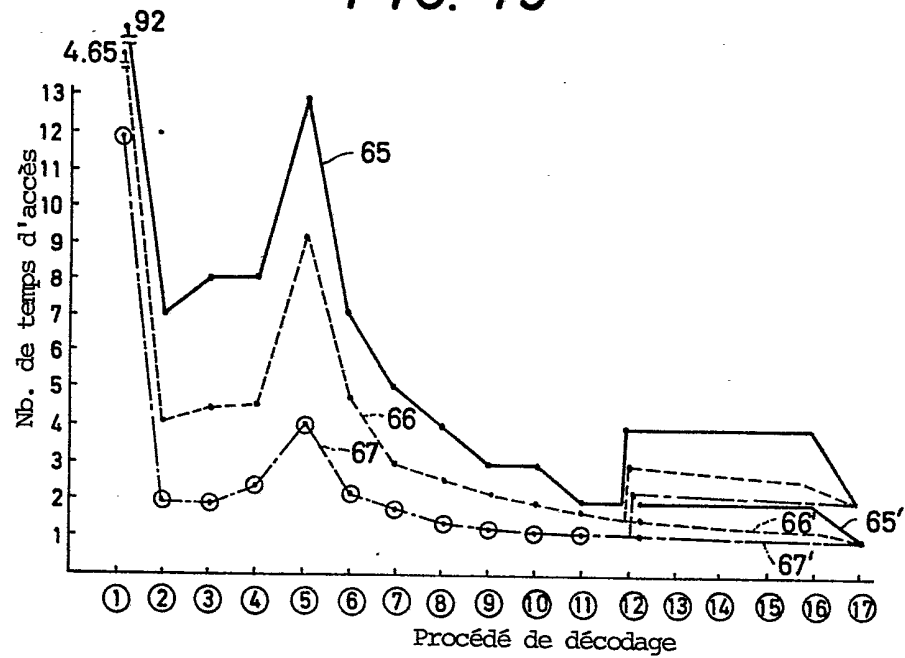


FIG. 16

		Capacité rel. déc. mém. à tables	Nb. relatif temps accès
Invention	Réalisation Fig. 7	1	1
	Réalisation Fig. 10	1,6	1 ~ 0,5
Ex. art antérieur des fig. 1 et 2		1,2	2
Ex. art antérieur de la fig. 3		3 ~ 100	0,5
Décodage passé sur la table de codes		0,9	12