

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Numéro de publication:

0 197 893
A1

(12)

DEMANDE DE BREVET EUROPEEN

(21) Numéro de dépôt: **86810154.4**(51) Int. Cl.4: **E05B 49/00**(22) Date de dépôt: **27.03.86**

(30) Priorité: **29.03.85 CH 1375/85**
14.05.85 FR 8507476

(43) Date de publication de la demande:
15.10.86 Bulletin 86/42

(84) Etats contractants désignés:
AT BE CH DE FR GB IT LI LU NL SE

(71) Demandeur: **Relhor S.A.**
74, rue du Locle
CH-2300 La Chaux-de-Fonds(CH)

(72) Inventeur: **Jolidon, François**
Terreaux 9
Ch-2300 La Chaux-de-Fonds(CH)
Inventeur: **Richard, Willy**
Convers 5
CH-2616 Renan(CH)

(74) Mandataire: **Caron, Gérard et al**
SMH Société Suisse de Microélectronique et
d'Horlogerie S.A. Département Brevets et
Licences Faubourg du Lac 6
CH-2501 Bienne(CH)

(54) **Dispositif de levée d'interdiction conditionnelle de la manoeuvre d'une serrure.**

(57) Dans ce dispositif, la serrure (5, 8) ne peut être manoeuvrée que si un verrou (11) est effacé. Ce dernier est entraîné par un moteur (22) commandé par un micro-processeur (30) qui est associé à une mémoire (32) dans laquelle sont stockées des fenêtres de levée d'interdiction de la manoeuvre de la serrure. Ces fenêtres sont constamment confrontées à l'heure officielle par le micro-processeur (30) qui ne lève l'interdiction en commandant le moteur 22 que si l'heure officielle se trouve à l'intérieur d'une fenêtre. Des intervalles d'inhibition primant les fenêtres peuvent intervenir pour les rendre inopérantes et empêcher la levée d'interdiction pendant ces intervalles. Un clavier (14) permet de modifier éventuellement après vérification d'un code d'autorisation, les fenêtres et les intervalles d'inhibition.

Application à la protection des coffres-forts, et autres enceintes protégées.

EP 0 197 893 A1

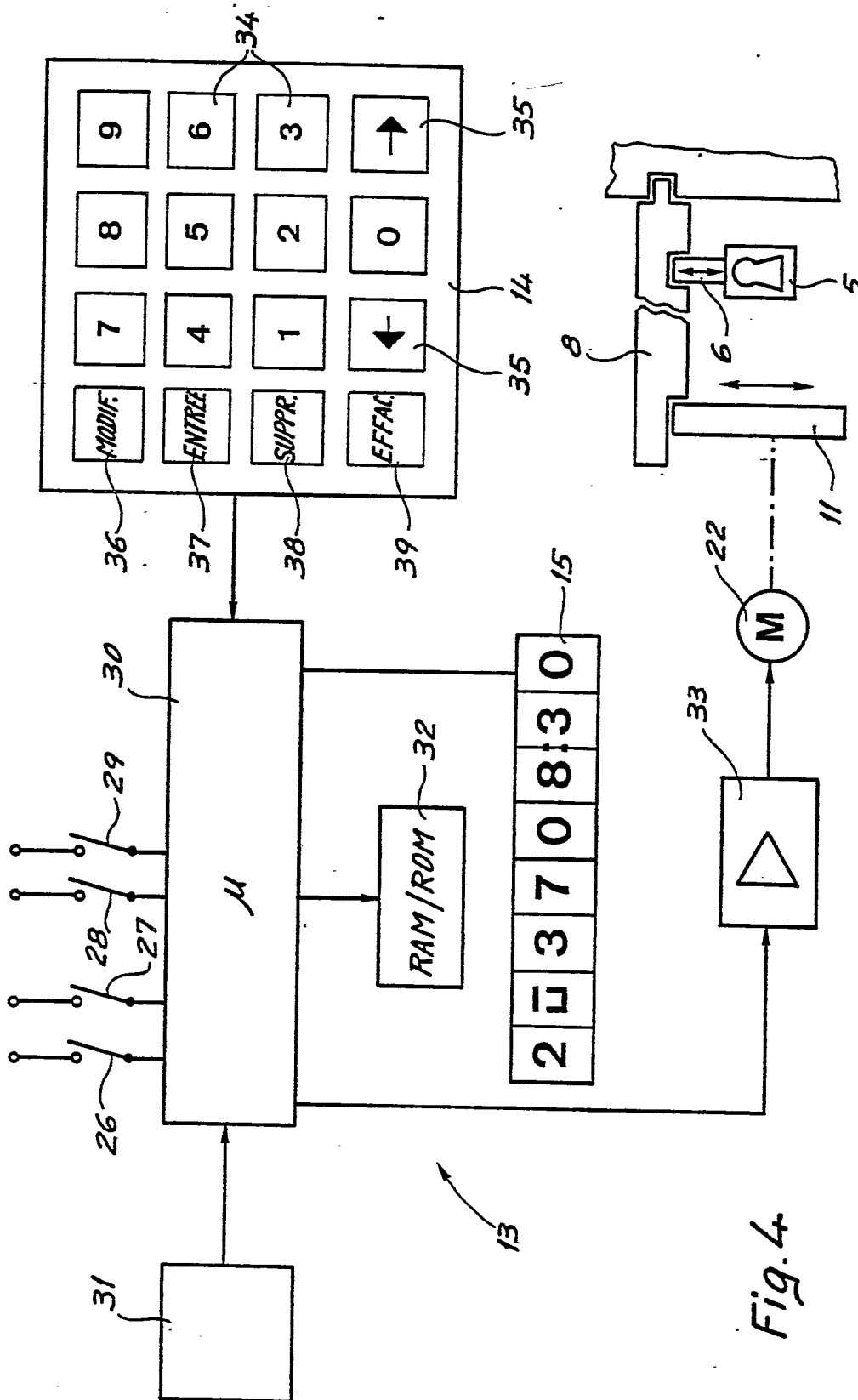


Fig. 4

DISPOSITIF DE LEVEE D'INTERDICTION CONDITIONNELLE DE LA MANOEUVRE D'UNE SERRURE

La présente invention est relative au domaine de la protection d'enceintes de sécurité et elle concerne, plus particulièrement, un dispositif de levée d'interdiction conditionnelle de la manoeuvre d'une serrure, notamment d'un coffre-fort, chambre forte ou autre enceinte protégée.

A l'heure actuelle, il est fréquent que l'on impose à la possibilité d'ouverture d'une enceinte protégée des fenêtres temporelles afin d'éviter, dans toute la mesure du possible, les violations criminelles de la protection et l'exposition à des dangers importants des personnels de service.

Ainsi, a-t-on déjà associé aux serrures des moyens de blocage qui en interdisent la manoeuvre pendant certaines périodes de la journée même pour les personnes qui possèdent la clé de la serrure ou le code d'ouverture si celle-ci est à commande plus complexe. Comme de cette manière, les périodes de levée d'interdiction sont réduites au minimum, les risques de violation sont évidemment plus faibles.

Le problème primordial que pose une telle installation est la détermination des fenêtres temporelles correspondant aux levées d'interdiction, qui est fait de préférence de manière automatique et sans que le personnel qui se sert quotidiennement de l'enceinte protégée, puisse modifier les données des fenêtres tout au moins dans le sens de l'élargissement de celles-ci. Cette impossibilité pour le personnel d'ouvrir l'enceinte protégée constitue pour lui une sécurité supplémentaire, car une attaque en dehors des fenêtres de levée d'interdiction peut ainsi être déjouée par l'impossibilité alors inhérente du système de pouvoir être ouvert. Même, l'injonction d'ouvrir l'enceinte protégée sous la menace des armes ne peut alors conduire à l'ouverture, car le personnel disposant de la clé ou du code d'ouverture n'est pas à même de provoquer l'ouverture.

Il est par ailleurs souhaitable que les fenêtres temporelles puissent être modifiées en fonction des circonstances telles que la modification des horaires d'ouverture des guichets d'une banque, l'instauration d'une nocturne de vente dans un magasin à grande surface, ou pour un autre établissement la période de fermeture en raison des congés, par exemple. Une telle adaptation doit naturellement être réalisée par des personnes autorisées qui pourront ou ne pourront pas être celles appartenant au personnel se servant habituellement de l'enceinte protégée.

C'est à ce problème de créer un système du type susindiqué que l'invention tente d'apporter une solution en fournissant un dispositif de levée d'interdiction adaptable facilement, d'un usage commode, et d'une grande flexibilité quant à ses possibilités d'installation sur des enceintes protégées de toute nature.

L'invention a donc pour objet un dispositif de levée d'interdiction conditionnelle de la manoeuvre d'une serrure, notamment d'un coffre-fort, chambre forte ou autre enceinte protégée, comprenant des moyens électro-mécanique de blocage qui sous la commande d'un circuit électronique sont capables d'imposer l'interdiction ou la levée d'interdiction de manoeuvre de ladite serrure, caractérisé en ce que ledit circuit électronique de commande comprend une mémoire dans laquelle est stockée un programme horaire destiné à être exécuté cycliquement et définissant au moins une fenêtre temporelle de levée d'interdiction et des moyens associés à cette mémoire pour comparer périodiquement l'heure officielle aux instants encadrant ladite fenêtre temporelle et des moyens pour commander lesdits moyens de blocage pour bloquer ou libérer ladite serrure lorsqu'il y a égalité entre l'heure officielle et lesdits instants.

Grâce à ces caractéristiques, les levées d'interdiction de la manoeuvre de la serrure sont déterminées en fonction de l'heure officielle de sorte que les fenêtres qui encadrent les levées d'interdiction peuvent être fixées simplement en introduisant dans le dispositif une heure d'ouverture et une heure de fermeture.

L'invention sera mieux comprise à l'aide de la description qui va suivre d'un exemple de réalisation, en faisant référence aux dessins annexés sur lesquels:

-la fig. 1 représente schématiquement la face avant d'un coffre-fort comportant l'ensemble de serrure suivant l'invention;

- la fig. 2 est une vue également schématique de la face intérieure de la porte de coffre-fort;

-la fig. 3 est une représentation schématique plus détaillée du dispositif de blocage de la serrure;

-la fig. 4 est un schéma simplifié du circuit électronique de commande de la serrure suivant l'invention;

-la fig. 5 est un diagramme représentant un exemple de programme hebdomadaire pouvant régir les périodes de levée d'interdiction de manoeuvre de la serrure;

-la fig. 6 est une représentation schématique du contenu de la mémoire RAM utilisée dans

le circuit électronique;

-la fig. 7 montre le contenu de plusieurs octets de la mémoire respectivement pour définir des données "courtes" et des données "longues";

-les fig. 8a à 8c montrent l'organigramme du programme principal exécuté par le micro-processeur de l'ensemble de serrure suivant l'invention;

-la fig. 9 représente le principe d'un sous-programme de comparaison;

-la fig. 10 est un organigramme d'un sous-programme de comparaison sur des données courtes;

-la fig. 11 est un organigramme d'un sous-programme de comparaison sur des données longues;

-les fig. 12a à 12e forment ensemble l'organigramme du sous-programme de "calcul de prochain événement";

-la fig. 13 représente l'organigramme d'un sous-programme permettant de déterminer la relation entre l'heure actuelle et le programme hebdomadaire préétabli;

-la fig. 14 représente l'organigramme d'un sous-programme de "recherche de début d'une prochaine fenêtre";

-la fig. 15 représente l'organigramme d'un sous-programme de "saut d'un jour";

-la fig. 16 représente l'organigramme d'un sous-programme de "recherche de la fin d'une fenêtre actuelle";

-la fig. 17 représente l'organigramme d'un sous-programme de "mise à jour de semaine relative";

-les fig. 18a et 18b représentent ensemble un organigramme de sous-programme de temporisation.

Dans la description qui va suivre, l'invention est explicitée à l'aide d'un exemple qui concerne spécifiquement un coffre-fort. Cependant, cet exemple n'est nullement limitatif, l'invention pouvant être appliquée chaque fois qu'il s'agit d'assurer la commande sous condition d'une serrure pour la protection d'une enceinte pouvant être obturée par un élément de fermeture.

Les fig. 1 et 2 représentent respectivement la face avant d'un coffre-fort et la face intérieure de la porte de ce dernier. L'ensemble de serrure de ce coffre-fort étant classique, il n'est pas utile d'en donner ici une description détaillée ou une représentation complète. Il suffit de noter que cet ensemble de serrure 1 comporte une tringlerie 2 munie de pènes 3 et pouvant être déplacée d'une position de verrouillage à une position de déverrouillage et inversement à l'aide d'un volant

4. Une serrure dont le coffre est représenté en 5 et dont le pêne est représenté en 6 peut être actionné à l'aide d'une clé classique à travers un trou de serrure 7, cette serrure pouvant bien entendu être aussi complexe qu'on peut le souhaiter dans le cadre de la sécurité à assurer vis-à-vis du contenu du coffre.

Le pêne 6 coopère avec une barre 8 de la tringlerie 2, cette barre étant ici montée coulissante dans le sens horizontal et comportant une crémaillère 9 avec laquelle coopère un pignon 10 monté sur le volant 4.

On conçoit que la manoeuvre de la tringlerie 2 à l'aide du volant 4 est interdite lorsque la serrure est fermée (pêne 6 dans sa position haute, comme représentée).

Le dispositif suivant l'invention permet d'opposer à la manoeuvre de cette tringlerie 2 un second élément de verrouillage 11 dont la commande détermine la levée d'interdiction sous condition de l'ouverture de la serrure du coffre-fort. En effet, bien que cela ne soit représenté que très sommairement à la fig. 2, on voit que l'élément de verrouillage 11 pouvant se déplacer verticalement et empêcher ainsi le coulisement dans le sens de l'ouverture de la tringlerie 2 en pénétrant dans une découpe 12 qui se trouve à l'arrière de la barre 8. Comme il sera expliqué par la suite, le second élément de verrouillage 11 est motorisé et commandé par un circuit électronique 13 auquel sont reliés d'une part un clavier de commande 14 et d'autre part un dispositif d'affichage 15, le clavier se trouvant à l'intérieur de la porte, alors que le dispositif d'affichage comprend deux champs d'affichage apparaissant respectivement de part et d'autre de celle-ci.

En se référant maintenant à la fig. 3, on va décrire, également très sommairement, un agencement possible de la partie mécanique du dispositif de blocage de la serrure suivant l'invention, les détails de réalisation apparaissant clairement au spécialiste à partir de la description qui va suivre de cette représentation.

L'élément de verrouillage 11 est réalisé sous la forme d'un coulisseau monté mobile axialement dans un guide 16 d'un bâti 17 ménagé sur la porte du coffre-fort. Ce dernier présente également un couloir 18 dans lequel coulisse la barre 8 de la tringlerie 2 et sur lequel débouche le guide 16 de manière que le coulisseau 11 puisse s'opposer au mouvement de recul de la barre 8 lorsqu'en sortant du guide 16, elle se place en travers du couloir 18.

Ce mouvement est commandé par un ensemble moteur 19 comprenant un moteur proprement dit 20, un levier 21 articulé sur le bâti 17 et une came 22 montée sur l'arbre de sortie du moteur 20

et pourvue d'une gorge de guidage excentrique 23 en forme d'arc de cercle, gorge dans laquelle coulisse une goupille 24 solidaire du levier 21. Le levier 21 est attaché par son extrémité libre à l'extrémité inférieure du coulisseau 11 et comme la goupille 24 se trouve située à la moitié du levier 21, on conçoit que si le moteur 20 est actionné, le levier peut soulever le coulisseau 11 pendant que la goupille se déplace dans la gorge de guidage 23 et que le mouvement inverse provoque la descente de ce coulisseau. Il est à noter que lorsque la barre 8 se trouve dans une position dans laquelle elle obture le guide 16, le coulisseau 11 peut être sollicité vers sa position de blocage grâce à la présence d'un ressort 25 qui se trouve monté autour de la queue du coulisseau 11 et qui est comprimé lorsque le moteur est entraîné dans le sens de la levée du coulisseau 11. Par conséquent, lorsque dans ces conditions, la barre retourne dans sa position de fermeture (vers la droite sur la figure), le coulisseau 11 est immédiatement amené à se placer en travers du couloir 18 pour s'opposer à tout nouveau mouvement dans le sens de l'ouverture de la barre 8.

L'ensemble que l'on vient de décrire comporte encore quatre interrupteurs 26 à 29 qui ont les fonctions suivantes:

-interrupteur 26: il est placé en dessous du guide 16 pour constater que le coulisseau 11 se trouve dans sa position inférieure;

-interrupteur 27: il coopère avec le levier 21 pour fournir un signal lorsque le levier 21 se trouve dans sa position supérieure; par conséquent, lorsque l'interrupteur 26 est fermé, le coulisseau se trouve en bas et lorsque l'interrupteur 27 est fermé, le coulisseau se trouve en haut.

-interrupteur 28: il constate une "volonté d'ouverture", c'est-à-dire qu'il est actionné par l'extrémité arrière de la barre 8 pour constater que le volant 4 a été déplacé dans le sens de l'ouverture, alors que le coulisseau 11 se trouve encore dans sa position haute;

-interrupteur 29: il constate que la barre 8 est effectivement dans sa position arrière (coffre ouvert).

La fig. 4 représente un schéma simplifié du circuit électronique qui est utilisé avec le dispositif mécanique qui vient d'être décrit. On reconnaît sur cette figure la barre 8, le coulisseau 11 entraîné par le moteur 22, le clavier 14 et le dispositif d'affichage 15. Ce dernier, ainsi que le clavier 14, constituent des périphériques d'un microproces-

seur 30 qui est également connecté à un circuit à base de temps 31, à une mémoire 32 du type RAM/ROM et à un circuit amplificateur 33 chargé de fournir l'énergie au moteur 22 lorsqu'une commande lui provient à cet effet du microprocesseur 30.

Le clavier 14 comporte les touches numériques 34 habituelles, des touches de décalage 35 pour permettre d'agir successivement sur les données apparaissant dans les cases du champ d'affichage, vers la gauche et vers la droite, une touche de modification 36, une touche de confirmation 37, une touche "suppression" 38 ainsi qu'une touche "effacement" 39. Le clavier 14 est destiné à permettre l'introduction des données de programmation dans le circuit électrique selon des procédures qui peuvent être effectuées soit par l'utilisateur quotidien du coffre-fort, soit par une personne autorisée, moyennant quoi cette dernière doit disposer d'un code secret, en principe inconnu de l'utilisateur quotidien. De telles procédures d'accès par un code d'utilisateur étant connues dans la technique des micro-ordinateurs, il n'est pas nécessaire d'en faire une description détaillée ici.

Le dispositif d'affichage 15 comporte huit champs d'affichage dont les quatre champs regroupés à droite sont destinés à l'indication de l'heure et dont les quatre champs de gauche indiquent respectivement de droite à gauche: le jour de la semaine (1 à 7), la semaine relative (0 à 3), ce concept étant expliqué par la suite, l'ouverture ou la fermeture du coffre (la condition d'ouverture étant représentée ici), et enfin l'indication du mode de fonctionnement instantané du circuit électronique. Il est à noter que ce dispositif d'affichage 15 est du type inter-actif et permet donc un dialogue entre l'opérateur du clavier 14 et le dispositif d'affichage 15, notamment au cours d'une modification de la programmation du dispositif suivant l'invention.

La fig. 5 représente un diagramme en fonction du temps illustrant un exemple de fonctionnement pouvant être exécuté par le dispositif suivant l'invention. La courbe a de ce diagramme représente un programme hebdomadaire de base qui est cyclique et qui revient donc en principe toutes les semaines. Dans le cas représenté, chaque jour présente deux fenêtres d'ouverture, respectivement entre 10 et 12 heures et entre 1400 et 1800 heures. Ce programme est préétabli par une personne autorisée disposant du code d'accès secret, l'utilisateur quotidien ne pouvant le modifier. Il est à noter que dans l'exemple qui va être décrit dans la

suite, on peut programmer chaque jour jusqu'à quatre de ces fenêtres d'ouverture. Ainsi, ce programme détermine au préalable les périodes de levée d'interdiction de l'ouverture de la serrure.

Cependant, cette levée d'interdiction peut être soumise à certaines autres conditions qui peuvent être imposées soit par la personne autorisée, soit par l'utilisateur quotidien, ce qui est illustré par les courbes b et c de la fig. 5. En examinant d'abord la courbe b, le programme de base hebdomadaire de la courbe a peut être inhibé pendant une durée allant de plusieurs heures à quelques semaines. Contrairement au programme de base, cette inhibition n'est pas cyclique, mais doit être programmée par la personne autorisée coup par coup et être automatiquement effacée après son exécution. L'inhibition dite imposée (puisque l'utilisateur quotidien ne peut l'effacer) peut être utile par exemple pour interdire l'ouverture pendant une période de vacances, un pont de congé associé à une journée fériée, etc.. Dans l'exemple décrit, l'inhibition peut être donnée en semaines "relatives" (qui sont de 0 à 3 en nombre), en jours (de 1 à 7), en heures et en minutes entre l'instant de début et l'instant de fin d'inhibition. Dans l'exemple de la fig. 5, on suppose que la période d'inhibition imposée commence le mardi à 8.30 heures et se termine le vendredi à 6.00 heures.

La courbe c représente l'inhibition "choisie", c'est-à-dire sélectionnable par l'utilisateur quotidien qui ne dispose pas du code d'accès secret. Cette inhibition peut être utile lorsque l'utilisateur doit s'absenter dans la journée pendant une ou plusieurs périodes, ce qui est le cas dans l'exemple représenté pour la journée du lundi. En effet, l'interdiction d'ouverture est établie entre 10.00 heures et 11.30 heures et de 16.30 heures à 18.00 heures.

La courbe d représente ainsi les périodes d'ouverture effectives ou périodes de levée d'interdiction qui sont hachurées pour plus de clarté.

Bien entendu, la fig. 5 ne représente qu'un seul exemple de détermination de la programmation de la levée d'interdiction, toute autre combinaison des périodes d'ouverture et de fermeture pouvant être choisies en fonction des besoins, soit par le programmeur autorisé, soit par l'utilisateur quotidien. Par ailleurs, en tout temps, si l'on est à l'intérieur de l'enceinte, l'ouverture et la fermeture immédiates peuvent être demandées par les deux types d'opérateurs.

Il est encore à noter que la programmation des instants d'ouverture et de fermeture ou de levée d'interdiction ou d'interdiction ne doit être confondue avec la programmation du micro-processeur 30 dont on va maintenant décrire le fonctionnement à l'aide des organigrammes des figures 8a à 18b.

Toutefois, il convient tout d'abord d'examiner le contenu de la mémoire 32 où tout au moins de la partie essentielle de cette mémoire qui contient à la fois les données de programmation de la levée d'interdiction comme décrites ci-dessus, que le programme régissant le fonctionnement du micro-processeur 30.

La fig. 6 représente schématiquement l'organisation de la partie RAM de la mémoire 32 dans laquelle sont stockées les données qui sont essentielles pour la mise en oeuvre de l'invention. Le programme hebdomadaire y est consigné dans 112 octets au maximum dans l'exemple représenté, ce programme étant utilisé cycliquement semaine après semaine. A chaque jour peuvent être affectées quatre fenêtres de levée d'interdiction, numérotées F_1 à F_4 , chaque fenêtre nécessitant pour sa définition deux octets pour l'ouverture et deux octets pour la fermeture.

Un autre champ de la mémoire est destiné à contenir des paramètres variables dont certains occupent trois octets et dont d'autres n'en occupent que deux.

La fig. 7 représente deux exemples de groupes d'octets utilisés pour cette organisation de la mémoire.

La partie supérieure de la fig. 7 représente le contenu de deux octets, appelés ci-après donnée "courte", utilisés pour la définition d'une ouverture ou d'une fermeture d'une fenêtre F_1 à F_4 , tandis que la partie inférieure représente un groupe de trois octets correspondants à une donnée "longue".

Le premier octet d'une donnée courte comporte sur quatre bits les unités des minutes (0 à 9), les quatre bits les plus significatifs étant affectés aux dizaines de minutes (0 à 5). Le deuxième octet comportant dans ses quatre bits les moins significatifs, les unités d'heures (0 à 9), les deux bits suivants étant affectés aux dizaines d'heures (0 à 2), les bits restants correspondant respectivement à des variables P et S pouvant admettre chacun deux états 0 et 1. Le bit P, lorsqu'il est à 1 signifie que la donnée courte en question est programmée et, lorsqu'il est à zéro, que cette donnée n'est pas programmée. Le bit S, lorsqu'il est à 1, signifie qu'après traitement de cette donnée, le coulisseau 11 (fig. 2 et 3) doit être descendu pour débloquer le passage de la barre 8, tandis que lorsque ce bit est à zéro, le mouvement inverse doit être déclenché.

Une donnée longue comporte dans ses deux premiers octets les mêmes informations que celles utilisées pour définir une donnée courte, tandis que le troisième octet de la donnée longue définit dans ses quatre bits les moins significatifs les jours de la semaine (0 à 6) et dans les quatre autres bits les "semaines relatives" (0 à 3).

Le concept "semaine relative" signifie les jours qui s'étendent d'aujourd'hui à aujourd'hui + 6 jours pour la première semaine numérotée 0, les jours de la semaine relative no 1 s'étendant d'aujourd'hui + 7 jours à aujourd'hui + 13 jours, etc.. En d'autres termes, la semaine relative peut commencer à une journée quelconque de la semaine du calendrier, c'est-à-dire par rapport au programme hebdomadaire défini dans la mémoire 32.

Dans les organigrammes qui vont être décrits ci-après, les données suivantes sont utilisées en tant que variables, ces données pouvant être stockées provisoirement dans la partie inférieure de la zone de mémoire apparaissant sur la fig. 6. Ainsi:

-PROG (JR, NE) représente la donnée du programme hebdomadaire de base correspondant à l'événement NE du jour JR (donnée courte), $0 \leq JR \leq 6$ et $0 \leq NE \leq 7$ dans l'exemple décrit.

- HRAC représente la donnée contenant l'heure actuelle, mise constamment à jour par une partie du programme du microprocesseur (donnée longue).

-NXTE représente normalement la donnée définissant l'instant du prochain événement, cette donnée étant le résultat d'un calcul définissant cet événement en fonction des contraintes ou conditions imposées, soit par le programmeur autorisé, soit par l'utilisateur quotidien comme autant de dérogations au programme hebdomadaire qui est défini au préalable. Il s'agit ici d'une donnée longue.

-AJT0 représente la donnée définissant le début d'une inhibition choisie par l'utilisateur quotidien - (AJT pour "ajout"). C'est donc le début d'une fermeture supplémentaire que l'utilisateur quotidien peut faire intervenir dans une fenêtre pour le jour actuel en cours (donnée courte).

- AJT1 représente la donnée définissant la fin d'une inhibition choisie à l'ouverture. C'est également une donnée courte.

-INH0 et INH1 représentent les données contenant respectivement le début et la fin d'une période d'inhibition imposée par le programmeur autorisé. Ce sont des données longues.

Pour expliquer le traitement des parties significatives des données longues et courtes, les organigrammes utilisent les désignations de variables suivantes:

-Mi (x) représente la partie "minutes" de la variable x, où x peut être l'une des données suivantes: HRAC, NXTE, AJT0, AJTA, INH0, INH1, PROG - (JR, NE).

-Hr (x) représente la partie "heures" de la variable x.

-P (x) représente le bit de la variable x qui indique si la donnée est programmée ou non.

-S (x) représente le bit de la variable x qui indique le sens du mouvement du coulisseau 11 associé à la donnée en question.

Les quatre parties de variable portent sur des données courtes x, et ces parties peuvent également être affectées aux deux premiers octets de chaque donnée longue y qui peut être l'une des données suivantes:

-HRAC, NXTE, INH0, INH1,

ces données étant complétées par:

-JR (y) qui représente la partie "jours"

-Sm (y) qui représente la partie "semaines".

Les fig. 8a à 8c représentent l'organigramme du programme principal mis en oeuvre dans le micro-processeur 30. Ce programme principal comporte un certain nombre de sous-programmes ainsi qu'une boucle de calcul qui se déroule dans la dernière phase de l'exécution de ce programme principal. Cette boucle de calcul comporte elle-même un certain nombre de sous-programmes qui seront tous décrits en détail ci-après.

Le programme principal est constitué essentiellement de quatre parties dont la première assure le calcul de l'heure (fig. 8a), dont la seconde a essentiellement pour but de confronter l'heure actuelle à la donnée NXTE définissant l'événement calculé auparavant, la troisième partie assurant la commande proprement dite du mouvement du coulisseau 11 en fonction des contraintes imposées par le programme, et la quatrième partie étant la boucle de calcul dont on vient de parler.

Le programme principal est exécuté toutes les demi-secondes sous la commande d'un signal qui est appliqué au micro-processeur 30 par la base de temps 31. Ce dernier peut être constitué par une chaîne de division classique pilotée par un oscillateur à quartz.

Lors de l'initialisation en 50 (fig. 8a), un test est tout d'abord effectué en 51 pour savoir si le contenu d'un registre de secondes SECS est égal à 119. On notera au passage que le programme fait usage d'un certain nombre de registres qui, de façon classique, font partie du micro-processeur 30.

Si le test est négatif, le contenu du registre est incrémenté d'une unité en 52 et le programme prend fin en 53 jusqu'au déclenchement de la demi-seconde suivante. Si, par contre, le test s'avère positif, le nombre zéro est placé dans le registre SECS (opération 54).

On comprend que le nombre examiné au cours du test 51, choisi ici à 119 (120, c'est-à-dire 2 minutes) est fonction de la fréquence à laquelle on souhaite faire exécuter le programme principal et elle peut être différente d'une fois toutes les deux minutes.

Le contenu du registre SECS étant établi à zéro, un test est effectué en 55 pour déterminer si le premier octet de la donnée HRAC contenu dans la mémoire 32 (unités et dizaines de minutes) est égal à 59. Si le test s'avère négatif, le contenu de l'octet en question est incrémenté d'une unité en 56 et le programme passe sur un sous-programme "comparaison longue" 57 (fig. 8b) dont une description détaillée sera donnée ci-après. Si le test 55 s'avère positif, le nombre 0 est placé dans l'octet précité en 58 et le programme passe sur un test 59 pour déterminer si le contenu du deuxième octet de la donnée HRAC contient un nombre d'heures égal à 23. Si le test s'avère négatif, le contenu de cet octet est incrémenté d'une unité en 60 et le programme passe également au sous-programme 57. Si le test s'avère positif, le nombre zéro est placé dans l'octet en question en 61. Cela signifie évidemment que l'on passe d'une journée à la journée suivante, car l'heure correspond alors à minuit.

Le programme passe ensuite par trois sous-programmes 62, 63 et 64 appelés "programme de mise à jour de semaines relatives" dont une description détaillée sera donnée ci-après (fig. 8b).

Cette mise à jour étant effectuée le cas échéant, le programme passe sur un test 65 pour déterminer si le contenu de la partie jours de l'octet 3 de la donnée HRAC est égal à 6. Si le test s'avère négatif, le contenu de cette partie de l'octet 3 est incrémenté d'une unité en 66 et le pro-

gramme passe sur le sous-programme 57. Par contre, si le test s'avère positif, zéro est placé dans la partie concernée de l'octet 3 (opération 67) et le programme passe également au sous-programme 57.

D'une façon générale, au cours du déroulement du programme dans son ensemble, des comparaisons doivent être effectuées entre des données horaires représentant les variables qui peuvent se présenter au fur et à mesure. Ces comparaisons peuvent être réalisées sur des données longues et des données courtes et peuvent donner lieu, en supposant que l'une des données est désignée par A et l'autre désignée par B, à quatre résultats qui sont illustrés sur la fig. 9, qui est un sous-programme général "C" de comparaison, le détail des sous-programmes correspondants CDC et CDL apparaissant respectivement sur les fig. 10 et 11. Comme il s'agit de données horaires, les quatre résultats peuvent être les suivants: A précède B, (C1), A et B sont synchrones - (C2), A succède à B (C3) et enfin B n'est pas programmé (C4), c'est-à-dire que son bit P est à zéro.

La comparaison de donnée courte (CDC) se déroule donc de la façon suivante (fig. 10). Après initialisation (CDC1), un test est effectué en CDC2 pour contrôler la valeur "1" du bit P de la donnée B. Si ce test s'avère négatif, B n'est pas programmé et on peut passer à l'opération suivante. Par contre si le test s'avère positif, un nouveau test est effectué en CDC3 pour déterminer si la valeur "heure" de la variable B est supérieure, égale ou inférieure à la valeur "heure" de la variable A. A défaut d'égalité, le même test est effectué sur les valeurs des minutes des deux variables (CDC4), ce dont il résulte les constatations C1 et C3. Si les valeurs des minutes des deux variables sont égales, on obtient le résultat C2.

S'il s'agit de variables correspondant à des données longues, il s'ajoute au sous-programme CDC deux tests complémentaires (sous-programme CDL) portant respectivement sur les valeurs de jours et de semaines des variables A et B, ces test étant indiqués par les références CDL5 et CDL6 sur la fig. 11, les autres opérations identiques au sous-programme CDC portant les références CDL1 à CDL4.

En revenant maintenant sur la fig. 8b on voit que le sous-programme 57 correspond à la comparaison longue CDL, selon la procédure de la figure 11, de la variable HRAC et de la variable NXTE qui est, exprimée en données horaires, l'heure à laquelle le prochain événement va avoir lieu, cette variable ayant été calculée auparavant. Elle dépend

donc du programme horaire de la courbe a de la figure 5 et des contraintes introduites par les inhibitions imposées et choisies, selon les courbes b et c de cette même figure.

Si, lors de l'exécution du sous-programme 57, il s'avère que la variable NXTE n'est pas programmée ou qu'elle est inférieure ou supérieure à la variable HRAC, le programme principal prend fin et le dispositif attend l'initialisation suivante du programme en retournant à l'opération d'initialisation 50 lorsqu'une demi seconde s'est écoulée. Par contre, lorsqu'il y a égalité entre les deux variables, le programme principal va exécuter sa troisième partie pour placer le coulisseau 11 dans la position exigée par les données obtenues après l'exécution du sous-programme 57.

Ainsi, un test est effectué en 68, pour déterminer à quelle valeur s'est établie le bit S de la variable NXTE. Si ce bit est à zéro, le circuit engendre un signal "fermé logiquement" en 69, ce qui ne veut pas encore dire que le moteur 20 entraînant le coulisseau 11 sera actionné dans le sens de la fermeture. En effet, il convient d'abord d'examiner dans quelle position se trouve le coulisseau. Après avoir passé un test en 70, dont l'utilité sera expliquée par la suite, et en admettant que la réponse à ce test est négative, un test est effectué pour déterminer dans quelle position se trouve réellement le coulisseau 11 en fonction des positions des interrupteurs 26 et 27. Si alors on constate que le coulisseau est ouvert, le moteur est déclenché selon l'opération 72 et le coulisseau est soulevé par le moteur 20 commandé par le microprocesseur à travers l'amplificateur 33. Le programme passe alors sur la boucle "calcul du prochain événement" 73. Si les tests 70 et 71 s'avèrent positifs, le programme passe également sur cette même boucle 73, par l'opération commune 74.

Si le test 68 s'avère affirmatif, les mêmes opérations se déroulent, mais cette fois-ci vis-à-vis de l'état d'ouverture du coulisseau 11, le programme passant par les opérations 75 à 78 qui aboutissent tous également sur la boucle 73.

On constate donc que lorsque les trois premières parties du programme principal ont été exécutées, le dispositif place le coulisseau dans la position exigée par les contraintes illustrées sur la fig. 5 en fonction de la variable NXTE précédemment calculée, après quoi, au cours de la quatrième partie, le dispositif va examiner quelle est la nature du prochain événement et en calculer exactement la variable NXTE suivante qui est alors stockée dans la mémoire 32 pour être utilisées lors de l'exécution du prochain sous-programme 57 conduisant à la comparaison avec l'heure actuelle.

On va maintenant décrire cette quatrième partie, c'est-à-dire la boucle de programme 73 servant à calculer le donnée NXTE traduisant les coordonnées horaires du prochain événement devant se dérouler. Cette boucle est initialisée en 79 (fig. 12a) et la première opération consiste à placer dans la mémoire 32 à l'emplacement de la donnée NXTE, la donnée HRAC de l'heure actuelle élaborée au cours de la première partie du programme principal (fig. 8a).

Ensuite est effectué un sous-programme CDL en 81 entre la donnée NXTE et le contenu des deux octets correspondant à la donnée INH1. Si, à la suite de cette comparaison, il s'avère que INH1 n'est pas programmé, est supérieur à la variable NXTE (correspondant à cet instant à l'heure actuelle) ou est égal à cette donnée, le programme passe sur un sous-programme suivant 82 appelé "localisation dans le programme hebdomadaire" - (LPH; voir fig. 12b).

Si, par contre, la comparaison du sous-programme 81 conduit à constater que INH1 est inférieur à NXTE, ce qui signifie que l'heure actuelle se situe, soit dans une période d'inhibition imposée, soit avant cette période, il s'agit de déterminer par le sous-programme CDL 82 qui est une comparaison longue entre INH0 et NXTE si l'heure actuelle est située dans la période d'inhibition imposée ou avant cette période. Si le sous-programme 83 détermine que l'on est encore avant la période d'inhibition, le programme passe sur le sous-programme 82. En revanche, s'il s'avère qu'il y a égalité entre les deux données, le programme passe sur l'opération 84 (fig. 12e) qui consiste à placer la donnée INH1, c'est-à-dire la fin de la période d'inhibition imposée dans les octets de la mémoire 32 correspondant à la variable NXTE chassant de ces octets la donnée HRAC qui y était précédemment stockée. On décrira plus loin les opérations suivant cette introduction dans les octets NXTE selon l'opération 84.

On va maintenant décrire le sous-programme LPH 82 en se référant plus particulièrement à la fig. 13.

Le sous-programme 82 a pour but de déterminer si la donnée contenue à cet instant dans les octets NXTE, selon l'opération 80 (fig. 12a) tombe dans une fenêtre d'ouverture du jour correspondant du programme hebdomadaire de base (fig. 5), et de déterminer s'il existe un événement du jour correspondant qui succède à cette donnée. Le sous-programme 82 est initialisé en 85 et la première opération 86 consiste à placer dans un registre JR du micro-processeur les bits du troisième octet de la donnée NXTE (qui correspondent donc au jour d'aujourd'hui), puis lors d'une

opération 87 de placer le nombre zéro dans une autre registre NE du micro-processeur. Les contenus des registres JR et NE sont alors utilisés comme pointeur des octets de la mémoire 32 dans lesquels est stocké le programme hebdomadaire - (courbe a de la fig. 5). Le contenu des octets désigné par le pointeur (en l'occurrence ce sont par exemple les deux octets d'ouverture de la fenêtre F, qui sont alors extraits de la mémoire) est soumis à un sous-programme de comparaison en 88 selon le processus de comparaison illustré sur la fig. 9.

Avant d'examiner les résultats possibles de la comparaison effectuée par le sous-programme 88, il est utile de signaler que la parité du nombre NE est représentative de l'état d'ouverture ou de fermeture d'une fenêtre, un numéro impair correspondant obligatoirement à l'ouverture et un numéro pair à une fermeture, étant entendu que dans l'exemple représenté, au maximum quatre ouvertures et fermetures successives peuvent se produire, le nombre NE des événements étant donc au maximum égal à huit.

Si la comparaison effectuée révèle que NXTE et PROG (JR, NE) sont égaux, le contenu du registre NE est incrémenté d'une unité en 89 et un test est effectué en 90 pour déterminer si NE est égal au nombre maximal des événements possibles programmés dans le programme hebdomadaire, ce nombre étant donc de huit dans l'exemple décrit. Si ce test s'avère négatif, le programme reboucle sur le sous-programme 88 pour extraire de la mémoire le deuxième événement du jour sur lequel pointent les registres JR et NE. Une nouvelle comparaison est alors effectuée sur NXTE, c'est-à-dire l'heure actuelle et PROG (JR, NE).

Si à la suite du test 90 il s'avère que NE est égal à 8, on se trouve devant un premier cas pouvant résulter du sous-programme 82. En effet, on saura alors que le coulisseau 11 se trouve dans sa position fermée et qu'aucun événement du programme hebdomadaire ne succède à l'heure du jour actuelle ce jour-là.

Si la comparaison effectuée par le sous-programme 88 indique que l'heure actuelle (NXTE) précède PROG (JR, NE), le programme effectue un nouveau test en 91 pour déterminer si NE est pair ou impair. Si NE est pair, on se trouve devant un deuxième cas, à savoir que le coulisseau est dans sa position fermée et qu'il y a un événement du programme qui va succéder à l'heure actuelle ce jour-là. Si, par contre, le test effectué en 91 s'avère positif, on se trouve devant un troisième cas correspondant à la position d'ouverture du coulisseau 11 et la programmation, ce jour-là, de la fin de la fenêtre en présence.

Si la comparaison effectuée par le sous-programme 88 aboutit à établir que l'heure actuelle succède à PROG (JR, NE), le programme continue comme précédemment décrit à propos de la constatation d'égalité des deux données correspondantes. Si, enfin, le sous-programme 88 établit que PROG (JR, NE) n'est pas programmé, le programme passe sur un test 92 pour examiner la parité du nombre NE. Si ce nombre est pair, on se trouve en présence du cas no 1 et dans le cas contraire on se trouve en présence d'un cas no 4 correspondant à la position d'ouverture du coulisseau 11, c'est-à-dire l'heure actuelle se trouve dans une fenêtre d'ouverture et cette fenêtre dure jusqu'à minuit du jour en cours. Ainsi, en se référant de nouveau à la fig. 12b, on voit que le sous-programme 82 donne lieu à quatre cas possibles qui vont être examinés successivement maintenant.

Si le sous-programme 82 conduit au cas no 1 ou no 2, le programme passe sur un sous-programme RDPF 93 (Recherche de Début de Prochaine Fenêtre) dont les opérations ont été détaillées sur la fig. 14 à laquelle on va se référer maintenant. Le sous-programme 93 est initialisé en 94 et la première opération consiste à placer dans le registre NE la valeur zéro (opération 95). Ensuite, un test est effectué pour déterminer si NE est égal au nombre maximum d'événements du jour (opération 96). Si le test 96 établit l'inégalité, le programme passe sur un test 97 pour déterminer le bit de programme P de la donnée PROG (JR, NE). Si ce bit est égal à un, un test est effectué en 98 pour déterminer la parité du nombre NE et si ce test s'avère négatif, les valeurs de minutes et d'heures de la donnée PROG (JR, NE) sont introduites dans les bits correspondants des trois octets du signal NXTE en repoussant les données de l'heure actuelle qui s'y trouvaient auparavant - (opération 99). Ensuite le programme reboucle sur le programme principal.

Si le test en 98 s'avère positif, le registre NE est incrémenté d'une unité en 100 et le programme reboucle sur le test 96 et les opérations précédemment décrites se répètent.

Si le résultat du test 96 est positif ou si celui du test 97 est négatif, le programme passe sur un sous-programme SDJ 101 appelé sous-programme de "saut d'un jour" que l'on va examiner maintenant en se référant à la fig. 15.

Ce sous-programme est appelé lorsque l'on cherche le prochain événement du programme hebdomadaire de base et que, ayant déjà atteint le dernier événement programmé du jour sur lequel on est en train de travailler, il faut passer au jour

suivant. Dans ce sous-programme on utilise une variable auxiliaire NC qui a pour but d'éviter, qu'en l'absence de programme on change plus de sept fois de jour.

Le sous-programme 101, après initialisation en 102, effectue un test 103 pour déterminer si le contenu du registre NC est égal à 7. Si c'est le cas, il s'agit d'une erreur qui est signalée sur le dispositif d'affichage et qui a pour conséquence la mise dans la position d'ouverture du coulisseau 11, ce qui est d'ailleurs le cas chaque fois qu'une erreur est constatée lors du déroulement du programme.

Si le test 103 s'avère négatif, le registre NC est incrémenté d'une unité (opération 104) et le nombre zéro est placé dans le registre NE. Puis, un test est effectué en 105 pour déterminer si le contenu du registre JR est égal à 6. Si ce test s'avère positif, le nombre zéro est placé dans le registre JR et dans le cas contraire, ce registre est incrémenté d'une unité (opérations 106 et 107). Puis, l'opération 108 consiste à transférer le contenu du registre JR dans les bits correspondant de la donnée NXTE de la mémoire 32. Ensuite, un test est effectué en 109 pour vérifier l'égalité entre le contenu du registre JR et le jour de la donnée HRAC. Si ce test s'avère négatif, le programme reboucle sur le test 97 du sous-programme 93 (fig. 14) et dans le cas contraire, le nombre de la donnée NXTE correspondant à la semaine est incrémenté d'une unité (opération 110) après quoi on repasse également sur le test 97.

A l'issue de l'opération 99 du sous-programme 93, le programme passe sur un sous-programme 111 (fig. 12b) qui consiste à effectuer une comparaison longue entre le contenu des octets de la donnée NXTE qui correspond alors non pas à l'heure actuelle, mais à une valeur calculée et la donnée INH0. Si cette comparaison détermine que INH0 n'est pas programmé ou si NXTE précède INH0, le programme passe sur l'opération 112 qui consiste à placer dans les bits P et S de la donnée NXTE la valeur 1, ce qui complète le calcul de cette donnée. Si, par contre, la comparaison établit l'égalité entre NXTE et INH0, ou si NXTE succède à INH0, une nouvelle comparaison longue est effectuée en 113 entre INH0 et INH1. Si INH1 succède à INH0, le programme passe sur l'opération 84 (fig. 12e). Dans les trois autres cas, il s'agit d'une erreur (opération 114) et un zéro est placé dans le bit PNXTTE.

On va maintenant examiner le déroulement des opérations lorsque, lors de l'exécution du sous-programme 82 (fig. 12b), les cas no 3 et 4 sont constatés en rappelant que le cas no 3 correspond à la position ouverte du coulisseau et la program-

mation de la fin de la fenêtre en cours, tandis que le cas no 4 correspond également à la position d'ouverture du coulisseau, la fenêtre étant établie jusqu'à minuit.

Lorsque le cas no 3 est constaté, le programme passe sur un test en 115 (fig. 12c), alors que si c'est le cas no 4, on effectue un test en 116, les tests 115 et 116 ayant pour but de déterminer si l'ordre d'ouverture existe. Si le test 115 s'avère négatif, un sous-programme 117 de comparaison courte est effectué entre la donnée NXTE qui est ici la donnée correspondant à l'heure actuelle et la donnée AJT1 qui est relative à une inhibition choisie. Si cette dernière donnée n'est pas programmée ou si elle est égale ou inférieure à la donnée NXTE, le programme reboucle sur le sous-programme 93 (fig. 12b). Si, par contre, AJT1 est supérieur à NXTE, le programme passe à un nouveau sous-programme 118 de comparaison courte entre la donnée PROG (JR, NE) (voir sous-programme 82) et la donnée AJT1. Au cas où AJT1 n'est pas programmée, si elle est égale à PROG (JR, NE) ou encore si AJT1 succède à PROG (JR, NE) le programme reboucle également sur le sous-programme 93. Par contre, si PROG (JR, NE) succède à AJT1, les valeurs de minutes et d'heures de cette donnée sont transférées dans les bits correspondant des octets NXTE de la mémoire 32 (opération 119). Après cette opération, le programme reboucle sur le sous-programme 111 de comparaison longue pour compléter les autres bits de la donnée NXTE lors de l'opération 112, ou pour déceler des erreurs (opération 114).

Le test effectué en 116 étant négatif, une comparaison courte est également effectuée sur les données NXTE et AJT1 (sous-programme 120), mais dans ce cas c'est seulement lorsque NXTE (heure actuelle) précède AJT1 que l'on passe à l'opération 119, les trois autres cas conduisant au rebouclage sur le sous-programme 93. Si les tests 115 et 116 s'avèrent positifs, les mêmes opérations de comparaison sont effectuées sur la donnée AJT0 conformément aux sous-programmes 121, 122 et 123 (fig. 12d).

Lorsque les sous-programmes 121 et 122 établissent soit que, AJT0 n'est pas programmé, soit que NXTE est supérieur à AJT0, soit encore que les deux données sont égales, le programme passe sur un sous-programme RFFA de Recherche Fin de Fenêtre Actuelle (Fig. 16). Le programme passe également sur ce sous-programme 124 à l'issue de l'exécution du sous-programme 123 lorsqu'il s'avère que AJT0 n'est pas programmé ou lorsque AJT0 est supérieur à PROG (JR, NE).

Si la comparaison établie lors du déroulement du sous-programme 122 aboutit à établir que AJTØ est supérieur à NXTE, les valeurs de minutes et d'heures de la donnée AJTØ sont placées les bits correspondants de octets de la donnée NXTE en remplacement des valeurs correspondantes de l'heure actuelle HRAC (Opération 125). De même, si, après le déroulement du sous-programme 123 il s'avère que AJTØ est supérieur à PROG (JR, NE), la valeur de minutes et d'heure de cette donnée sont placées dans les bits correspondants de la donnée NXTE de la mémoire 32 (opération 126).

Le sous-programme RFFA 126 a pour but de rechercher la fin de la fenêtre actuelle, pour permettre à la fin de cette fenêtre de placer dans le bit "S" du signal NXTE, la valeur 0 ayant pour but de provoquer le mouvement du coulisseau 11 dans le sens de la fermeture.

Le sous-programme 124 (fig. 16) se déroule pratiquement entièrement comme le sous-programme 93 conformément aux opérations 94 à 101 déjà décrites à propos de la fig. 14. La différence consiste en ce que, à l'issue du déroulement du sous-programme 101 (fig. 15), c'est-à-dire lorsque le test 109 s'avère négatif ou que la mise à jour des bits du signal NXTE correspondant au nombre de semaines, est effectuée selon l'opération 110, le sous-programme 124 effectue un test 127 pour déterminer si le bit P de la donnée PROG, (JR NE) est égal à 1, étant rappelé que cette donnée a été extraite de la mémoire en fonction du contenu des registres JR et NE. Si ce test 127 s'avère positif, deux tests successifs 128 et 129 sont effectués sur les valeurs d'heures et de minutes de la donnée PROG (JR, NE) pour vérifier si ces valeurs sont égales à zéro. Si les tests 128 et 129 s'avèrent négatifs, des valeurs zéro sont introduites dans les bits correspondants des octets de la donnée NXTE, conformément à l'opération 130, après quoi le programme rebranche sur le programme 73 de calcul du prochain événement (fig. 12d). Si les tests 128 et 129 sont tous les deux positifs, cela veut dire que, selon un cas particulier, on a programmé une fenêtre d'ouverture qui chevauche minuit. Il faut alors rechercher l'événement suivant de la nouvelle journée, de sorte que l'on incrémente la valeur NE selon l'opération 100.

Les opérations 127 à 130 ont donc pour but d'éviter que, lorsque l'ouverture est programmée pour chevaucher minuit, le coulisseau ne soit pas amené successivement dans la position de fermeture et dans la position d'ouverture, mais reste au contraire dans la position d'ouverture, malgré le fait qu'en réalité, à minuit, par convention, le coulisseau 11 doit toujours être dans sa position de fermeture.

Si, après le déroulement du sous-programme 124, aucune erreur n'est constatée, le programme - (fig. 12d) passe sur un sous-programme de comparaison longue 131 pour vérifier si la valeur actuelle de la donnée NXTE ne correspond pas à une donnée INHØ, c'est-à-dire avec un début d'inhibition imposée.

Si la donnée INHØ n'est pas programmée ou s'il succède à NXTE ou encore s'il est égal à INHØ, des valeurs de 1 et 0 sont respectivement placées dans les bits P et S de la donnée NXTE - (opération 132), après quoi le calcul de NXTE s'achève.

Si, par contre, la donnée NXTE s'avère succéder à INHØ, cette donnée est placée dans les octets de NXTE (opération 133), après quoi les mêmes valeurs de bits sont placées dans les bits P et S de NXTE selon l'opération 132. La même opération est effectuée si, après un test 134 fait après constatation d'une erreur dans le sous-programme 101 se déroulant dans le cadre de l'exécution du programme 124 (fig. 16), on vérifie si le bit P de la donnée INHØ est égal à 1. Si tel n'est pas le cas, 0 est placé dans ce bit et on constate une erreur.

En revenant sur la fig. 12a, en cas d'égalité entre NXTE et INHØ au cours du déroulement du sous-programme 83, NXTE se trouve dans une période d'inhibition. Le coulisseau 11 se trouve alors dans sa position fermée et il faut rechercher le début de la première fenêtre qui suit la fin de l'inhibition. Le programme se poursuit alors de la façon suivante (fig. 12e).

L'opération 84 déjà mentionnée consiste à placer dans les octets NXTE la valeur INH1, après quoi est exécuté le sous-programme 82 pour localiser cette donnée dans le programme hebdomadaire.

Si le sous-programme 82 aboutit au cas no 1 ou au cas no 2, on va exécuter le sous-programme 93 pour rechercher le début de la fenêtre consécutive. Si les cas no 3 ou no 4 se présentent, des valeurs 1 sont respectivement placées dans les bits P et S de la donnée NXTE et le calcul du prochain événement prend fin. Si le sous-programme 93 aboutit à la constatation d'une erreur, la valeur zéro est placée dans le bit P de la donnée NXTE et une erreur est constatée (opérations 136 et 137).

On va maintenant revenir sur le programme principal et plus précisément sur les sous-programmes 62, 63, 64 apparaissant sur la fig. 8b. Il a déjà été indiqué que les inhibitions imposées utilisent une notion de semaine relative dont la définition a été donnée. Etant donné cette relativité, la variable qui l'a décrite varie avec la progression du temps,

de sorte que, lorsque l'on passe d'un jour à l'autre, il faut mettre à jour les données qui contiennent l'indication de la semaine relative, à savoir les données NXTE, INH0 et INH1. C'est précisément le rôle des sous-programmes MJSR 62, 63 et 64 qui sont les mêmes et que l'on n'a représentés qu'une seule fois sur la fig. 17. Après initialisation en 138, ce sous-programme vérifie tout d'abord si la donnée en question est programmée par le test 139. S'il n'est pas programmé, le sous-programme prend fin. S'il est programmé, on vérifie au cours du test 140 si la valeur de jour de la donnée HRAC correspond à la valeur de jour de la donnée examinée. Si elle ne correspond pas, le sous-programme prend fin. S'il correspond, on vérifie au cours du test 141 si la valeur de semaine de la donnée examinée est égale à zéro. Si c'est le cas, la valeur zéro est placée dans le bit P de la donnée examinée. Par contre, si ce n'est pas le cas, la valeur de semaine de la donnée examinée est décrétementée d'une unité (opérations 142 et 143, respectivement) après quoi, le sous-programme en question prend fin et les opérations rebranchent sur le programme principal.

On va maintenant décrire les fig. 18a et 18b qui sont relatives à la temporisation à l'ouverture du coulisseau 11. L'algorithme correspondant est exécuté chaque demi-seconde après le calcul de l'heure (fig. 8a).

Dans la description qui précède des algorithmes régissant le programme du micro-processeur 30, on a utilisé plusieurs notions d'ouverture et de fermeture (levée d'interdiction ou interdiction) qui sont les suivantes:

1. L'ouverture au sens du programme. Se présente lorsque le temps actuel HRAC se trouve entre les limites d'une fenêtre d'ouverture du programme;

2. L'ouverture théorique. Est l'ouverture au sens du programme, éventuellement restreinte par une inhibition imposée. Au sens théorique, l'appareil est ouvert si l'heure actuelle se trouve entre les limites d'une fenêtre d'ouverture et ne se trouve pas entre les limites de la période d'inhibition imposée, éventuellement programmée.

3. L'ouverture logique. A l'extérieur des fenêtres d'ouverture théorique, l'appareil est logiquement fermé; à l'intérieur des fenêtres d'ouverture théorique, l'appareil peut être logiquement ouvert ou logiquement fermé, soit par action manuelle, soit par le jeu des inhibitions choisies.

4. L'ouverture physique. Correspond à l'état du coulisseau 11 déplacé par le moteur. Dans un dispositif suivant l'invention dépourvu de

système de temporisation, l'ouverture physique correspond à l'ouverture logique. Cependant il est possible de prévoir une temporisation et, dans ce cas, l'ouverture logique définit la période pendant laquelle un signal externe de requête d'ouverture est pris en considération et donne lieu, après une période d'attente programmable, à une ouverture physique d'une durée standard de trois minutes par exemple.

Le signal de requête peut être présenté à n'importe quel instant, signal qui peut être éventuellement associé à un code spécial connu des personnes autorisées seulement. Si l'ouverture est ainsi demandée, l'algorithme des fig. 18a et 18b impose une durée limitée à la possibilité d'ouverture, encore que cette ouverture ne se présente qu'après la période d'attente mentionnée ci-dessus. L'algorithme de temporisation des fig. 18a et 18b consiste, après une initialisation en 144, à examiner d'abord une transition du signal de requête à l'aide des tests 145, 146 et 147. Ces tests consistent à examiner l'état d'un drapeau FR qui est établi à zéro après le test 146 (opération 148) ou à un (149).

Après l'opération 148, un test est effectué en 150 pour déterminer s'il y a erreur. Si ce test est négatif, on va examiner en 151 s'il y a ouverture logique. Dans le cas positif, un test est effectué pour examiner si la variable TEMP est égale à zéro, cette variable étant la valeur de consigne correspondant à la durée de temporisation qui est ici de trois minutes. Si la variable TEMP est égale à zéro, le moteur 20 est commandé dans le sens de l'ouverture (opération 153), après quoi la variable TEMP reçoit la valeur 3. Si les tests 151 et 152 s'avèrent négatifs, le moteur 20 est commandé dans le sens de la fermeture (opération 154) qui reboucle également sur l'opération 155 d'ajustement de la variable TEMP. Alors le programme se poursuit par la mise à jour d'une variable TEMP0 à sa valeur initiale de 120 demi-secondes exécutées lors de l'opération 156 apparaissant sur la fig. 18b.

Après l'opération 149, qui consiste à établir un 1 pour le drapeau FR, on examine s'il y a erreur en 157, après quoi on vérifie si la variable TEMP0 est égale à zéro. Dans le cas négatif, la variable est décrétementée d'une unité en 159, après quoi, cette variable est de nouveau confrontée à la valeur zéro dans le test 160.

Si ce test s'avère positif, on vérifie, lors du test 161, si la variable TEMP1 est égale à zéro. Dans le cas négatif, cette valeur est décrétementée d'une unité (opération 162), après quoi on vérifie de nou-

veau, lors du test 163, si cette variable est égale à zéro. Dans le cas positif, le moteur 20 est commandé dans le sens de la fermeture (opération 164) et le programme reboucle sur l'opération 156. Dans le cas contraire, un nouveau test en 165 est effectué sur la variable TEMP1 pour vérifier s'il est égal à 3. Dans le cas négatif, le programme reboucle sur l'opération 156. Dans le cas contraire, le moteur 20 est commandé dans le sens de l'ouverture (opération 166). Enfin, si le test 161 s'avère positif, le programme reboucle sur le programme principal.

Revendications

1. Dispositif de levée d'interdiction conditionnelle de la manoeuvre d'une serrure (5, 8), notamment d'un coffre-fort, chambre forte ou autre enceinte protégée, comprenant des moyens électromécaniques de blocage (11, 19) qui sous la commande d'un circuit électronique (14, 30 à 33) sont capables d'imposer l'interdiction ou la levée d'interdiction de manoeuvre de ladite serrure, caractérisé en ce que ledit circuit électronique de commande (14, 30 à 33) comporte une mémoire - (32) dans laquelle est stockée un programme horaire destiné à être exécuté cycliquement et définissant au moins une fenêtre temporelle de levée d'interdiction (F_1 à F_n) des moyens associés à cette mémoire pour comparer périodiquement l'heure officielle (HRAC) aux instants (NXTE) encadrant ladite fenêtre temporelle et de moyens (30, 33) pour commander lesdits moyens de blocage pour bloquer, respectivement libérer ladite serrure lorsqu'il y a égalité entre l'heure officielle et lesdits instants.

2. Dispositif suivant la revendication 1, caractérisé en ce que ledit circuit électronique comporte un microprocesseur (30) programmé pour donner l'heure officielle (HRAC) et pour effectuer ladite comparaison à des intervalles réguliers.

3. Dispositif suivant la revendication 2, caractérisé en ce que chacun desdits instants est défini dans ladite mémoire (32) à l'aide d'une donnée (NXTE) comprenant des valeurs numériques de minutes, d'heures et le cas échéant de jour et de semaine, une information (P) établissant si cette donnée entre dans le programme horaire ou non et une information (S) définissant dans quel sens les moyens de blocage doivent être commandés, à savoir dans le sens d'une interdiction ou dans le sens d'une levée d'interdiction de la manoeuvre de la serrure (5).

4. Dispositif suivant la revendication 3, caractérisé

en ce que ledit microprocesseur (30) est programmé pour, après chaque constatation d'égalité entre l'heure officielle (HRAC) et un instant (NXTE) du programme horaire cyclique, extraire de la mémoire (32), la donnée correspondant à l'instant suivant de ce programme horaire, et pour vérifier l'égalité de cette nouvelle donnée et l'heure officielle (HRAC) au rythme desdits intervalles réguliers.

5. Dispositif suivant l'une quelconque des revendications précédentes, caractérisé en ce que les données stockées dans ladite mémoire (32) comportent également des données définissant des intervalles de temps au cours desquels la levée d'interdiction prévue par ledit programme horaire cyclique est inhibée (INH0, INH1, AJT0, AJT1).

6. Dispositif suivant la revendication 5 lorsqu'elle dépend de l'une quelconque des revendications 2 à 4, caractérisé en ce que ledit microprocesseur - (30) est également programmé pour, lors de chaque opération de comparaison, vérifier si l'instant suivant prévu par ledit programme horaire cyclique tombe au cours d'un intervalle correspondant à une inhibition de la levée d'interdiction.

7. Dispositif suivant la revendication 6, caractérisé en ce que ledit microprocesseur (30) est connecté à un dispositif de communication interactif (14, 15) et en ce qu'il est programmé pour permettre l'introduction à posteriori des données définissant les intervalles d'inhibition (INH0, INH1, AJT0, AJT1).

8. Dispositif suivant la revendication 7, caractérisé en ce que ledit microprocesseur (30) est programmé pour n'admettre l'introduction à posteriori des données d'intervalles d'inhibition que par l'intermédiaire d'un code d'autorisation.

9. Dispositif suivant l'une quelconque des revendications 1 à 8, caractérisé en ce que ledit programme horaire cyclique est hebdomadaire.

10. Dispositif suivant l'une quelconque des revendications précédentes, caractérisé en ce qu'il comporte des moyens pour différer l'action sur lesdits moyens de blocage (11, 19) pendant un intervalle de temps prédéterminé s'écoulant après constatation de l'égalité entre l'heure officielle et un instant correspondant à une levée d'interdiction de la manoeuvre de la serrure.

11. Dispositif suivant l'une quelconque des revendications 2 à 10, caractérisé en ce que ledit microprocesseur (30) est programmé pour déterminer le sens de commande desdits moyens de blocage - (30, 33) vers une interdiction ou une levée d'interdiction de la manoeuvre de la serrure en fonction du nombre d'instants s'étant présentés durant un intervalle de temps prédéterminé.

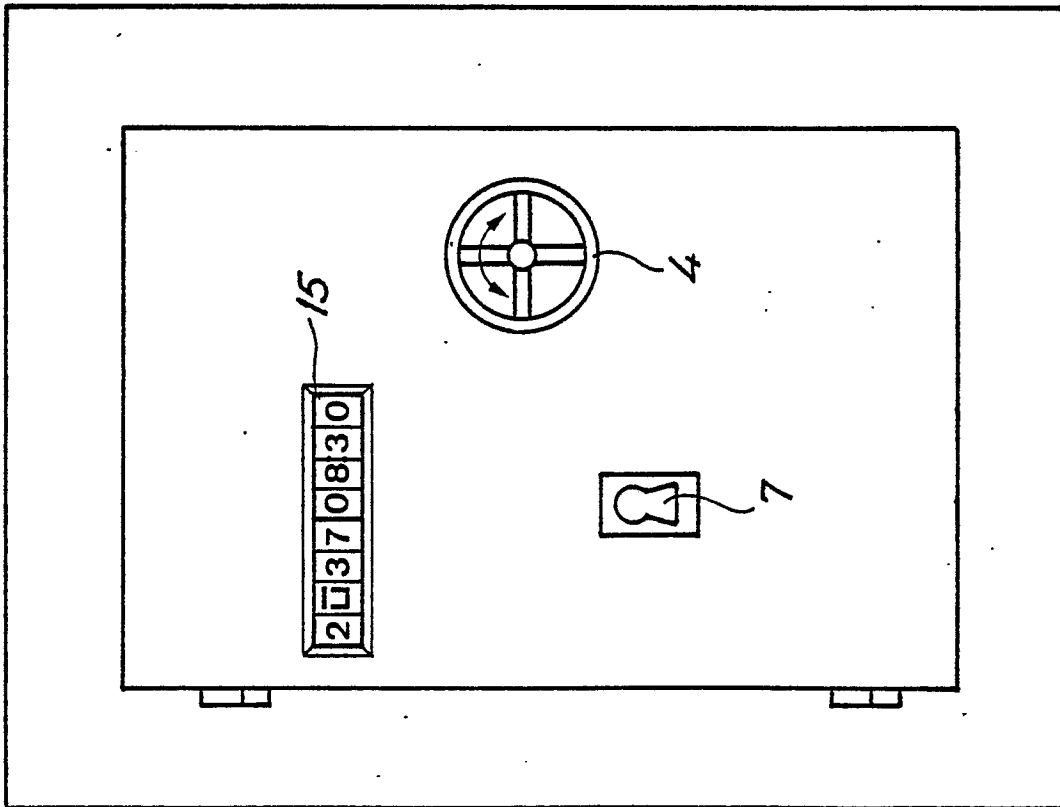


Fig. 1

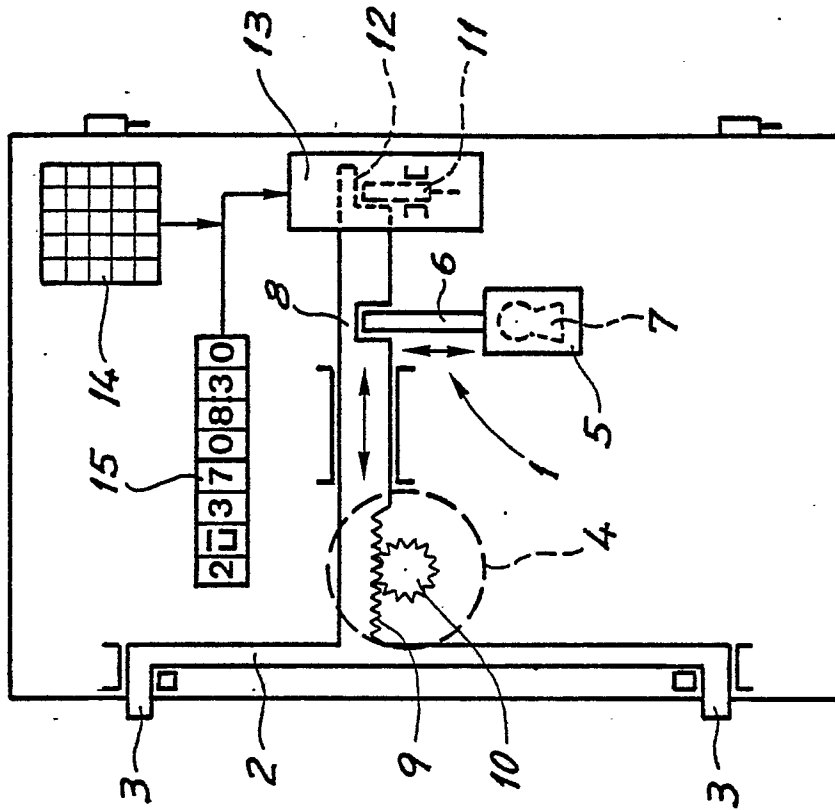
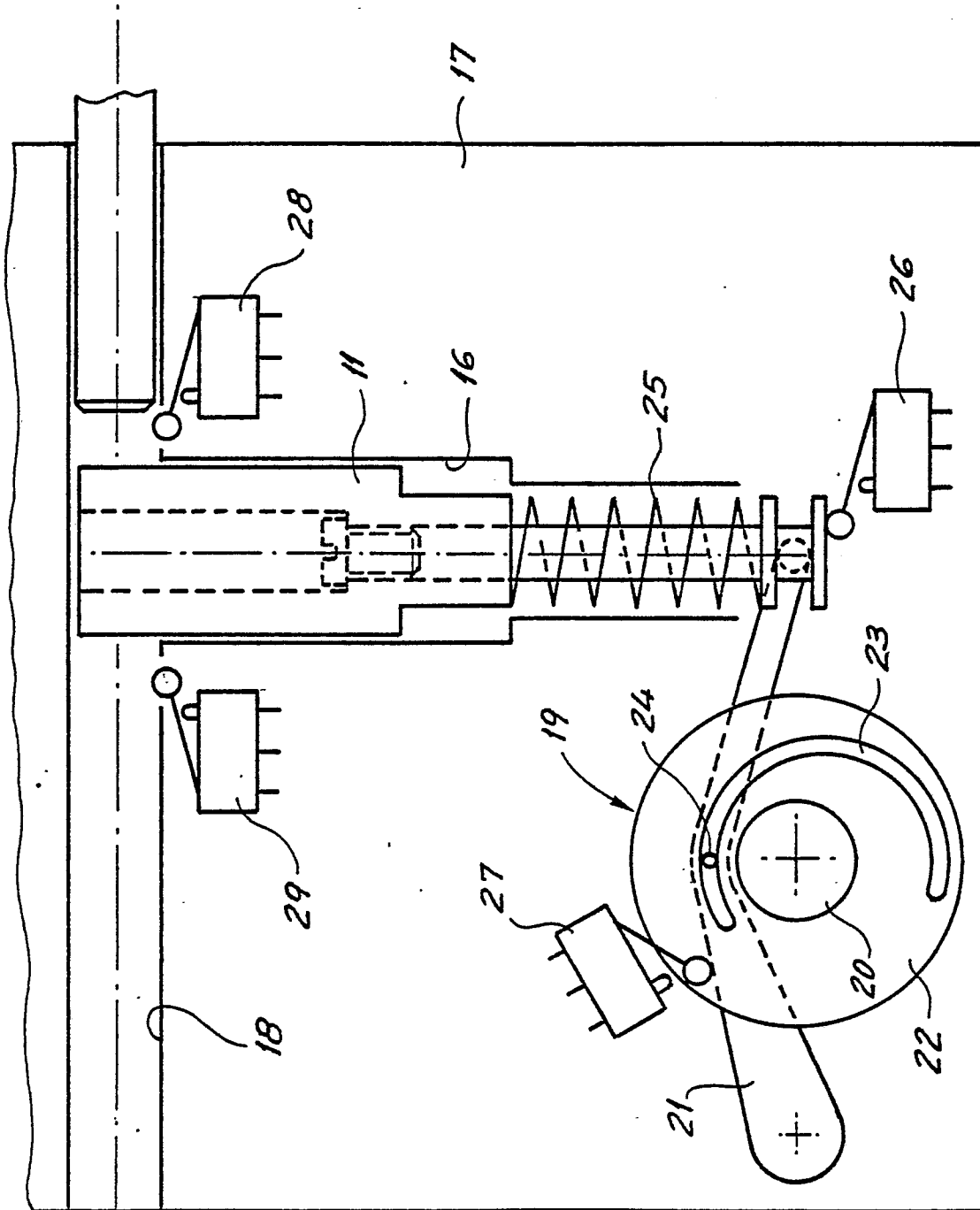


Fig. 2

Fig. 3



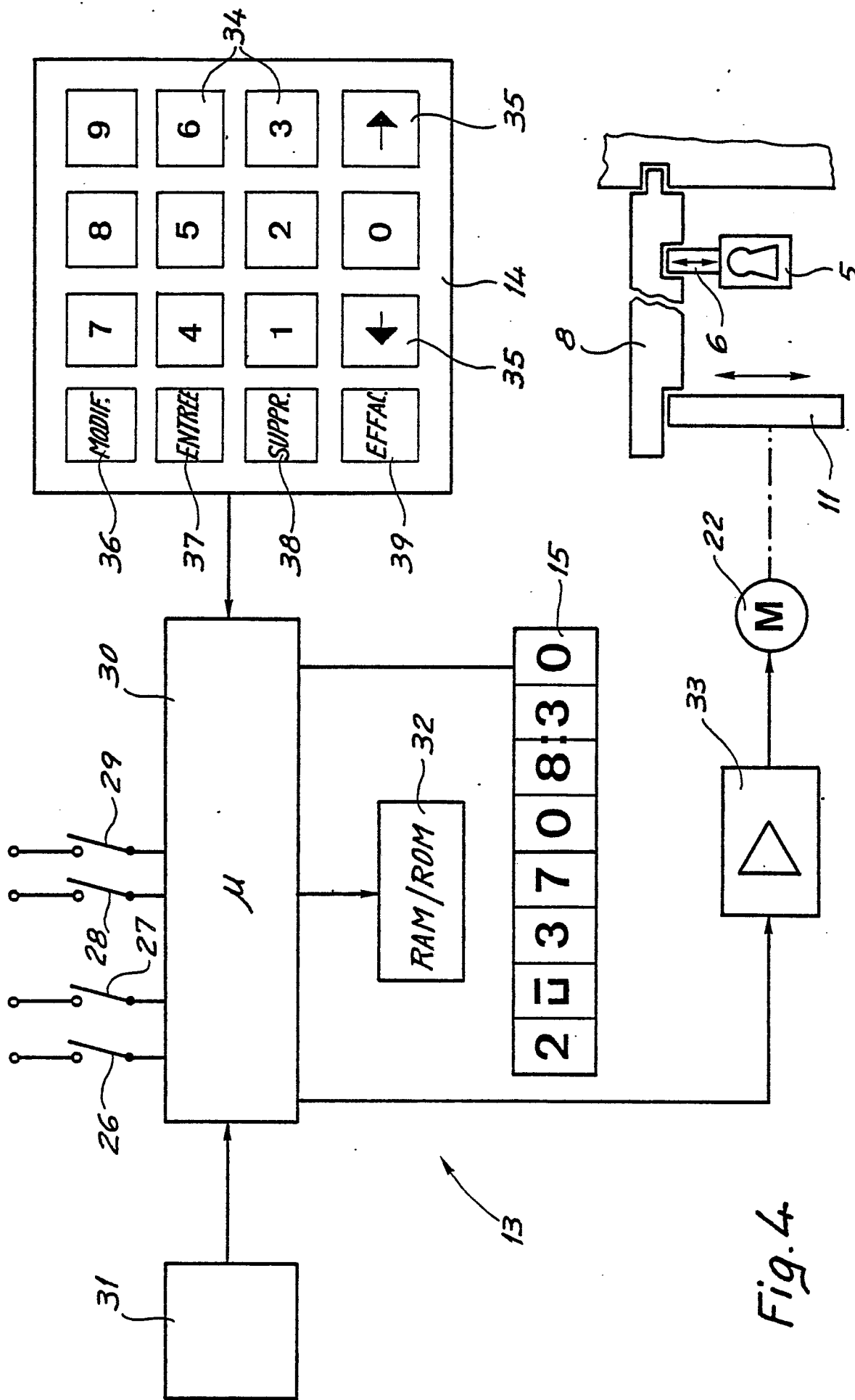


Fig. 4

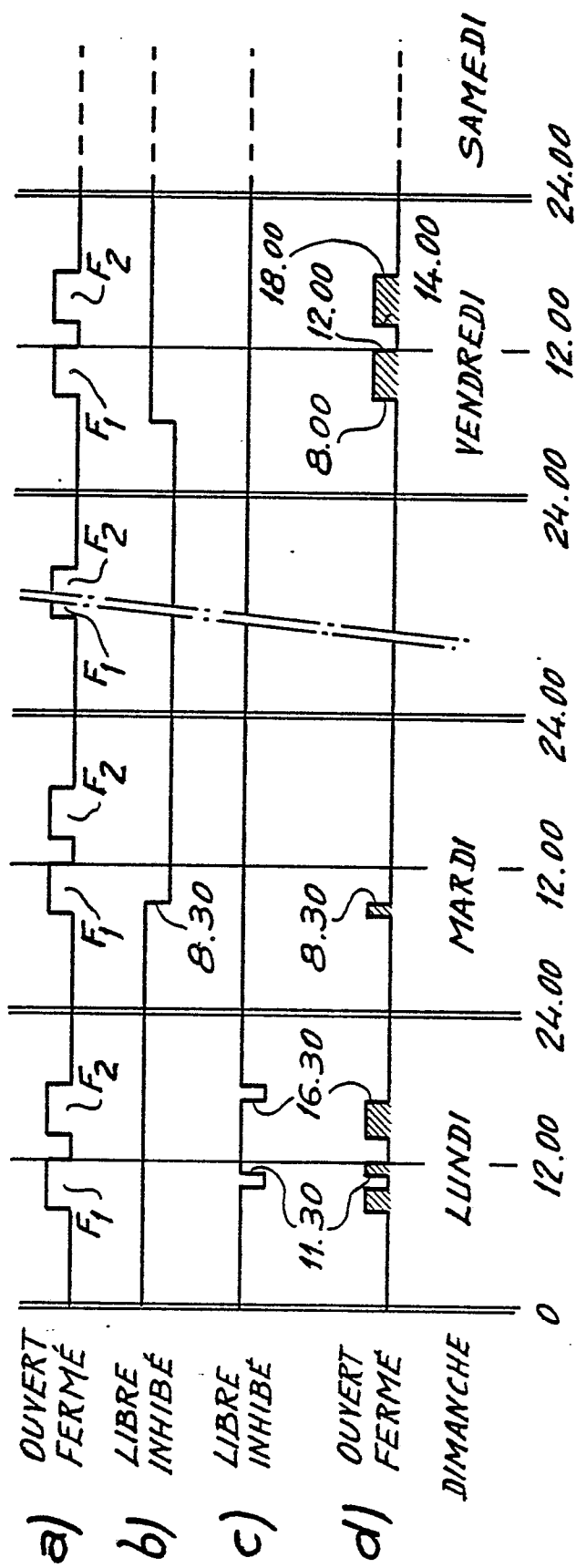


Fig. 5

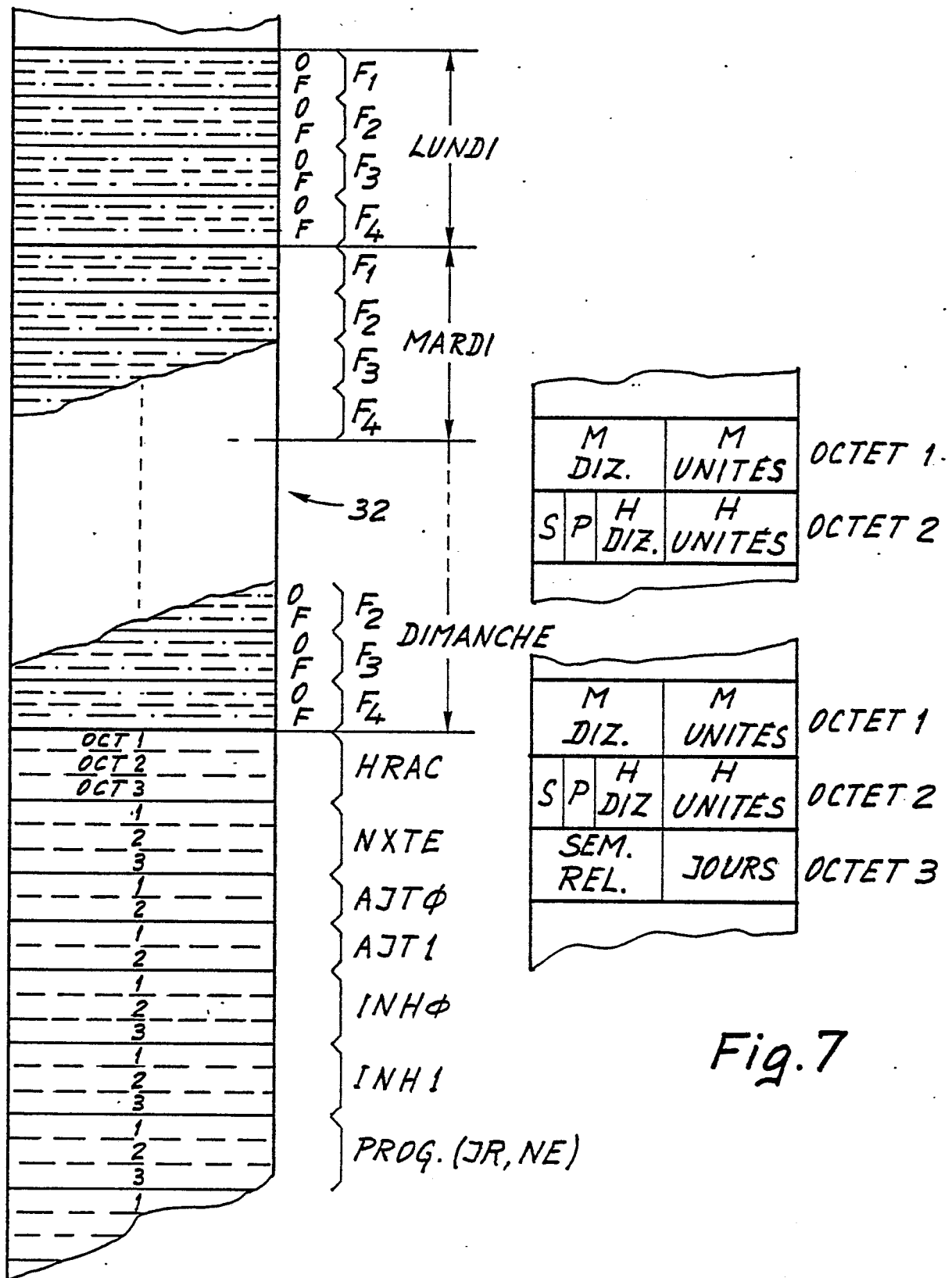


Fig.7

Fig.6

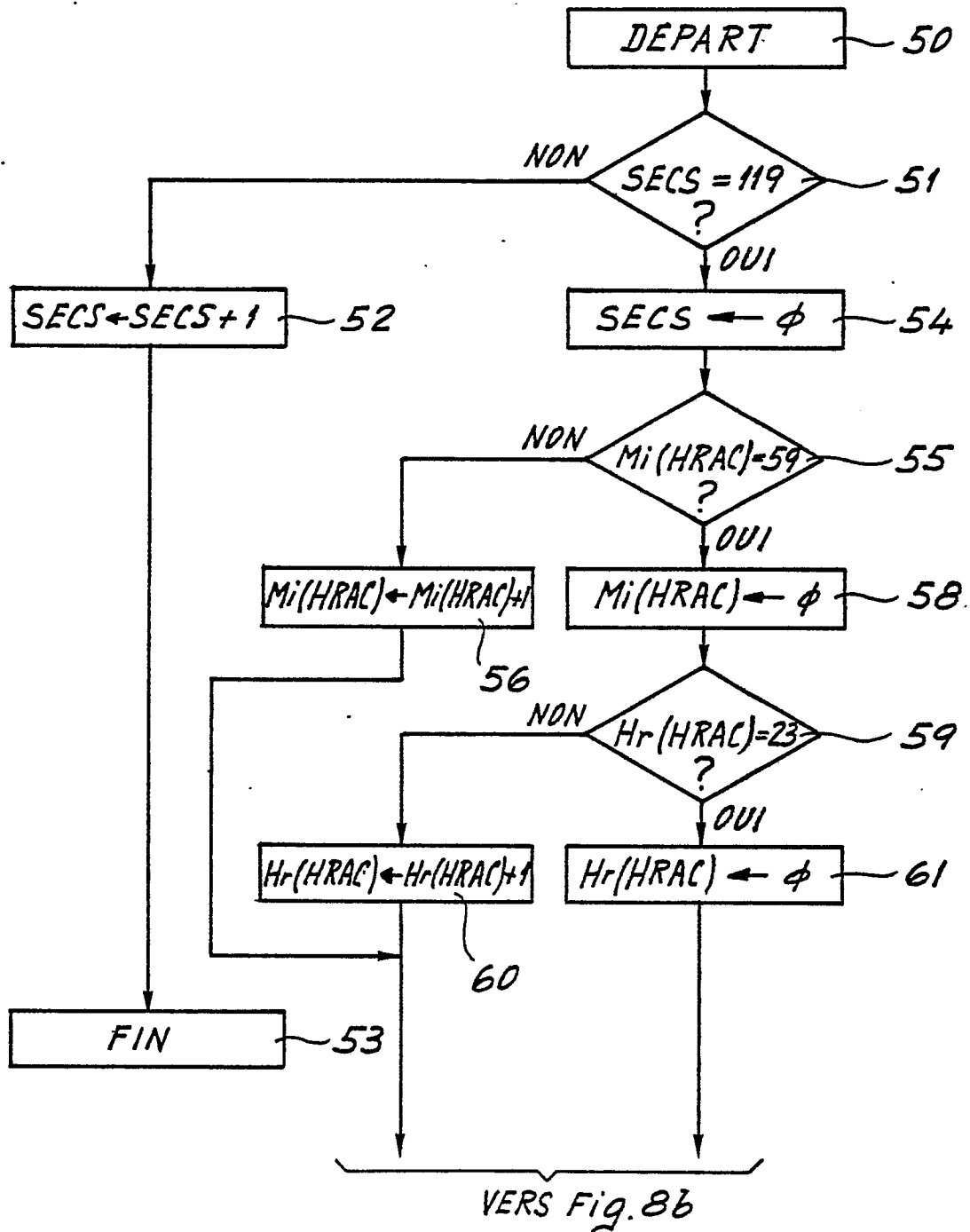


Fig. 8a

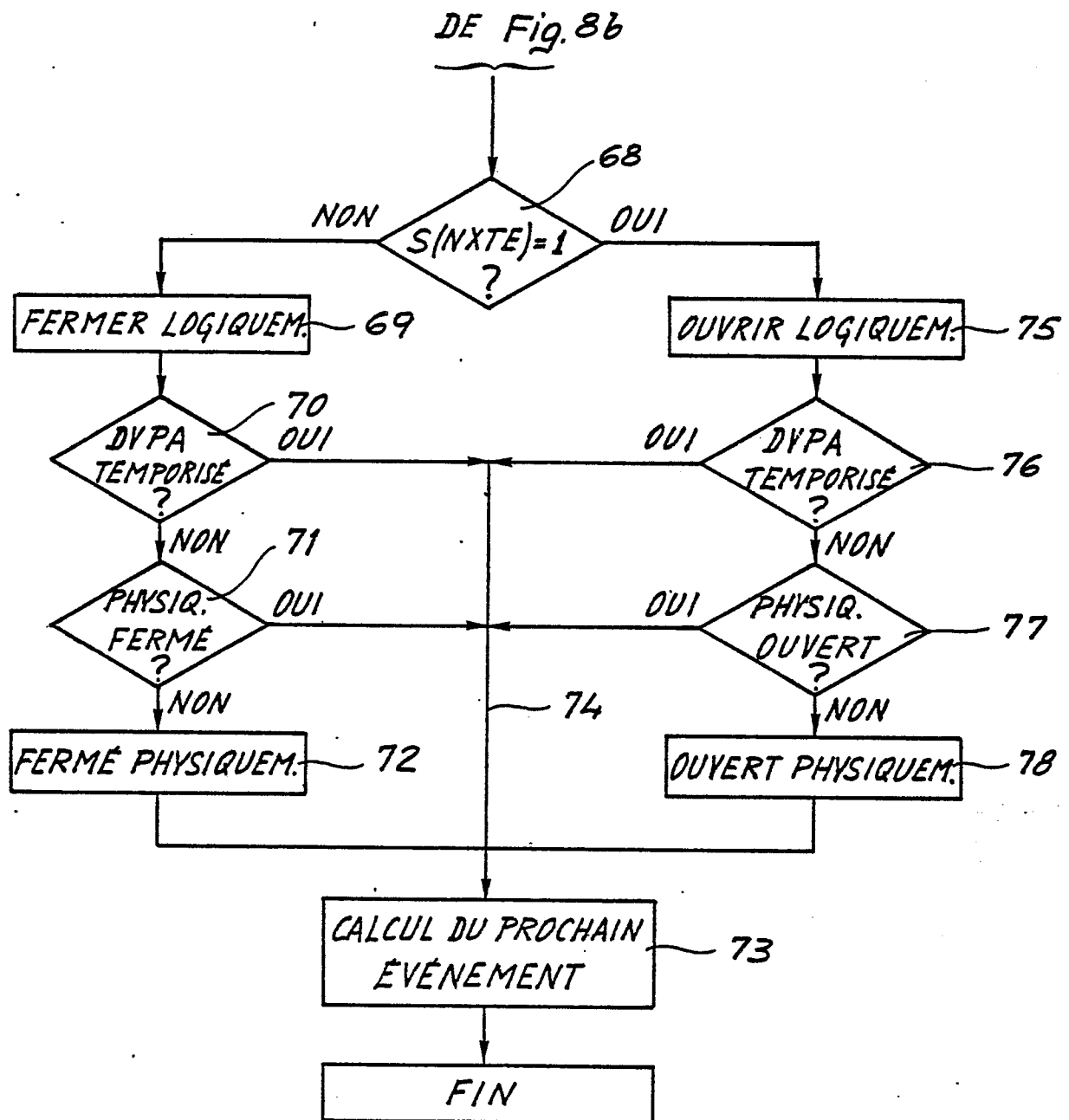
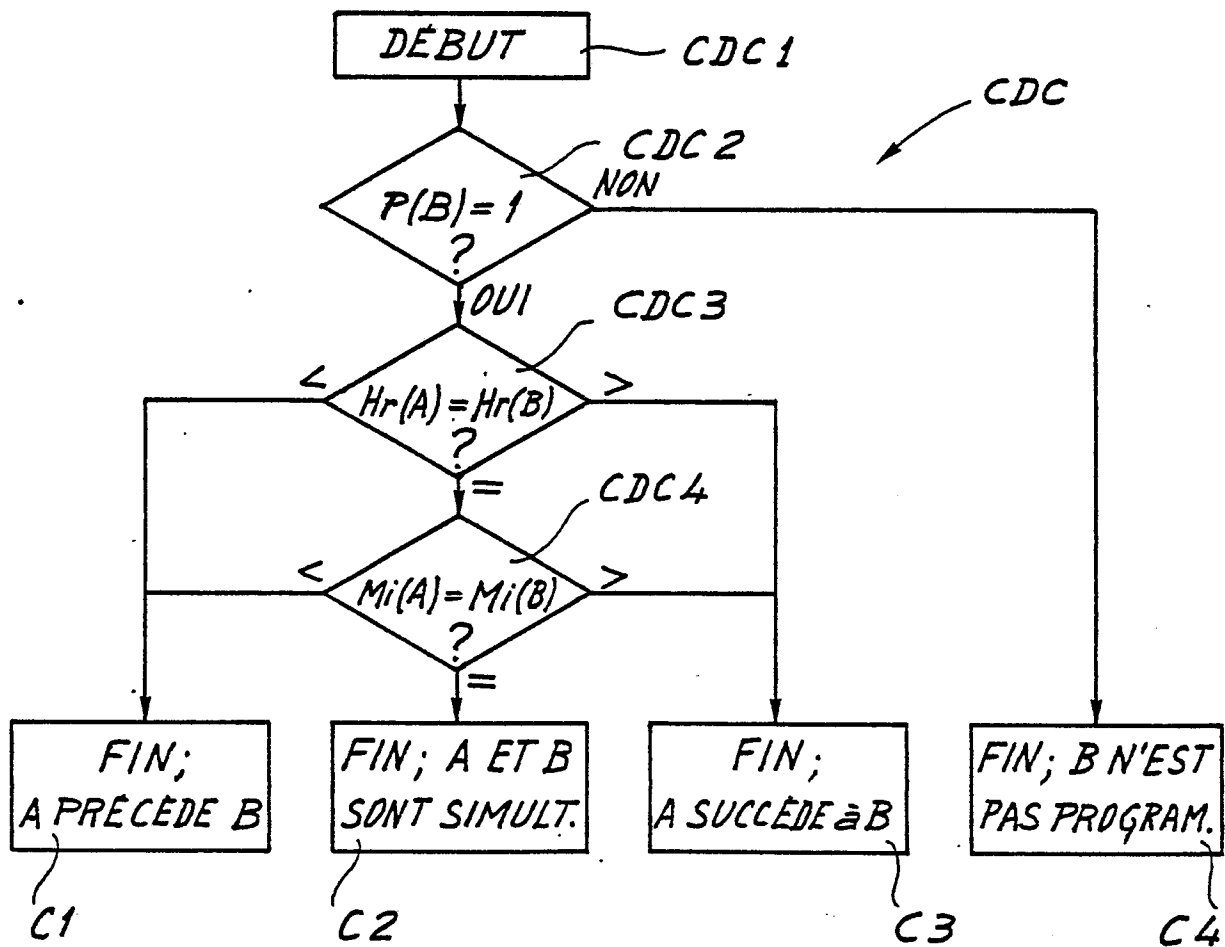
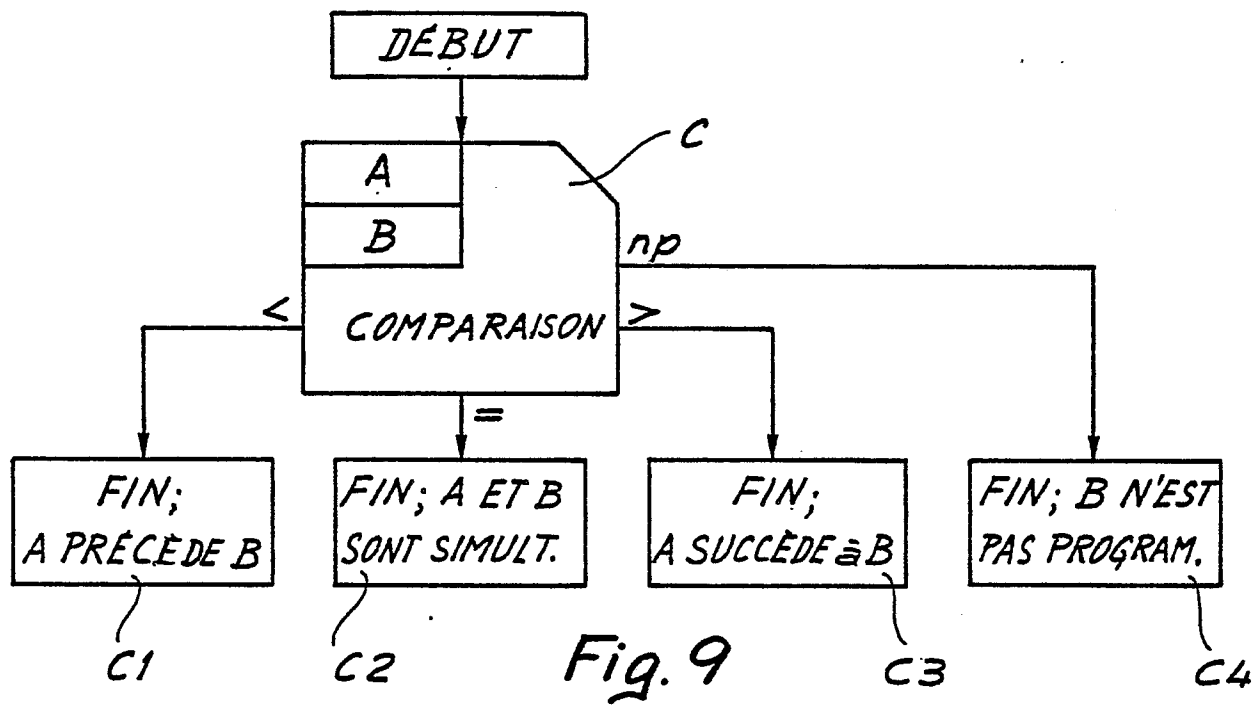


Fig. 8c



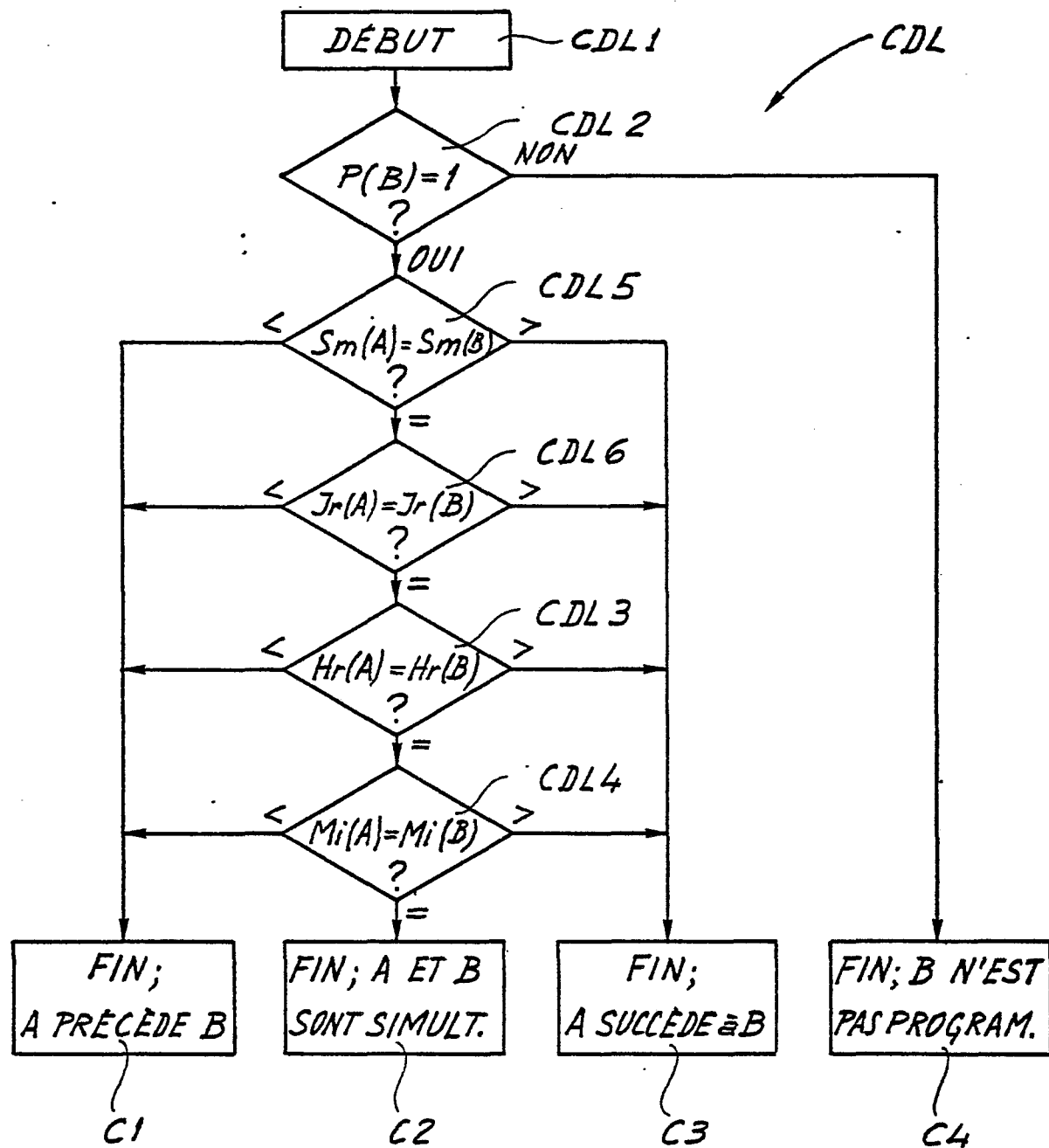


Fig. 11

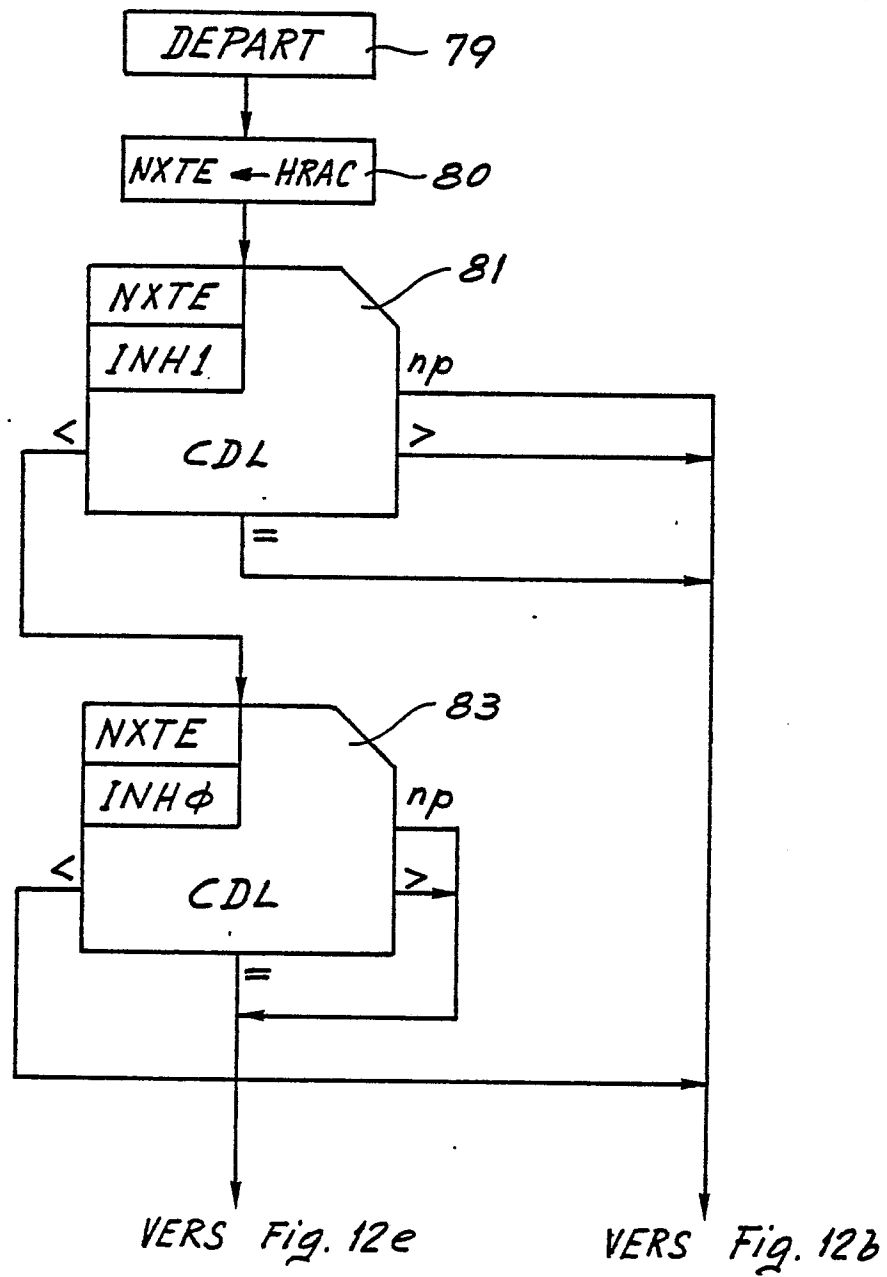
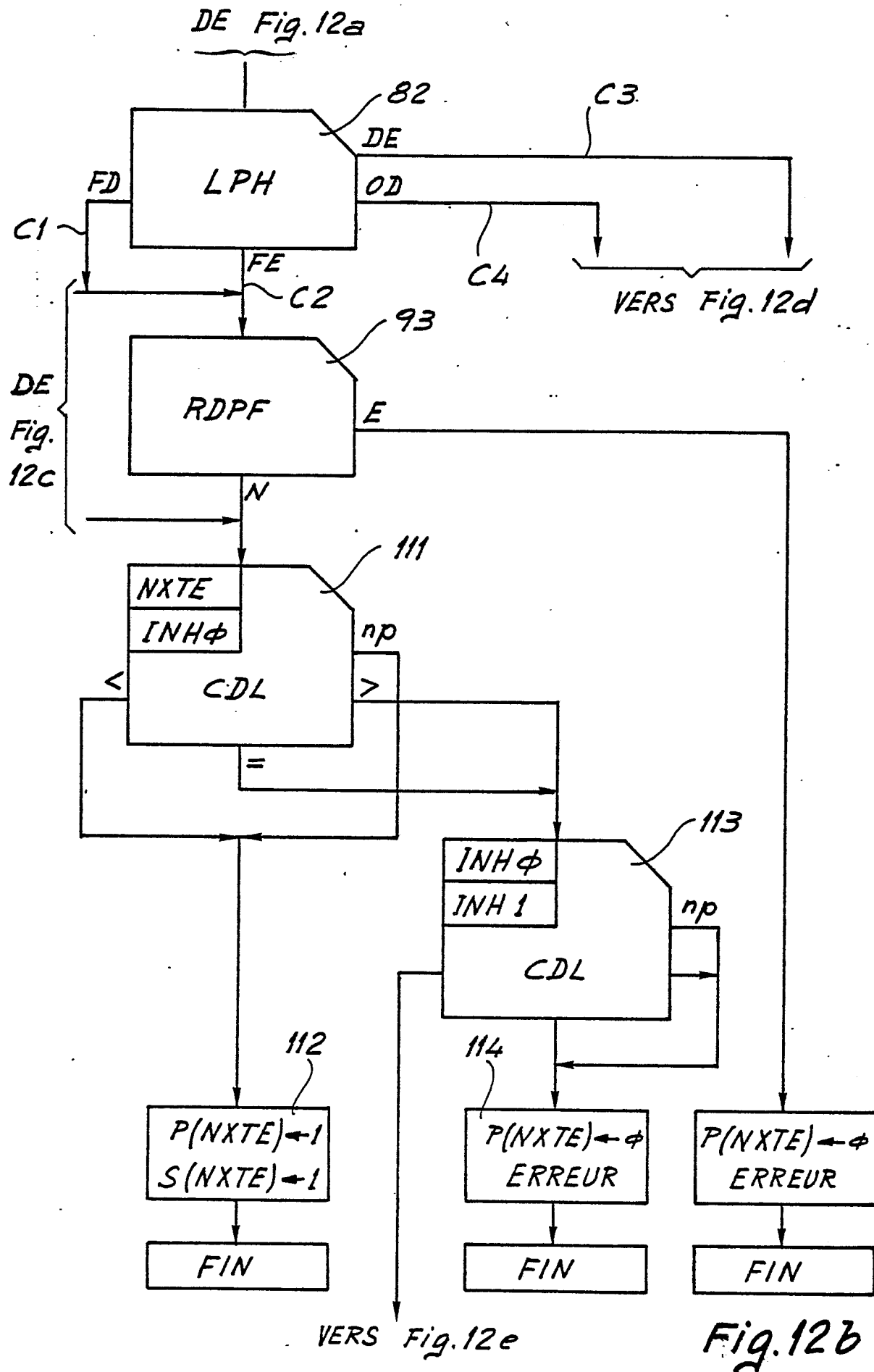


Fig. 12a



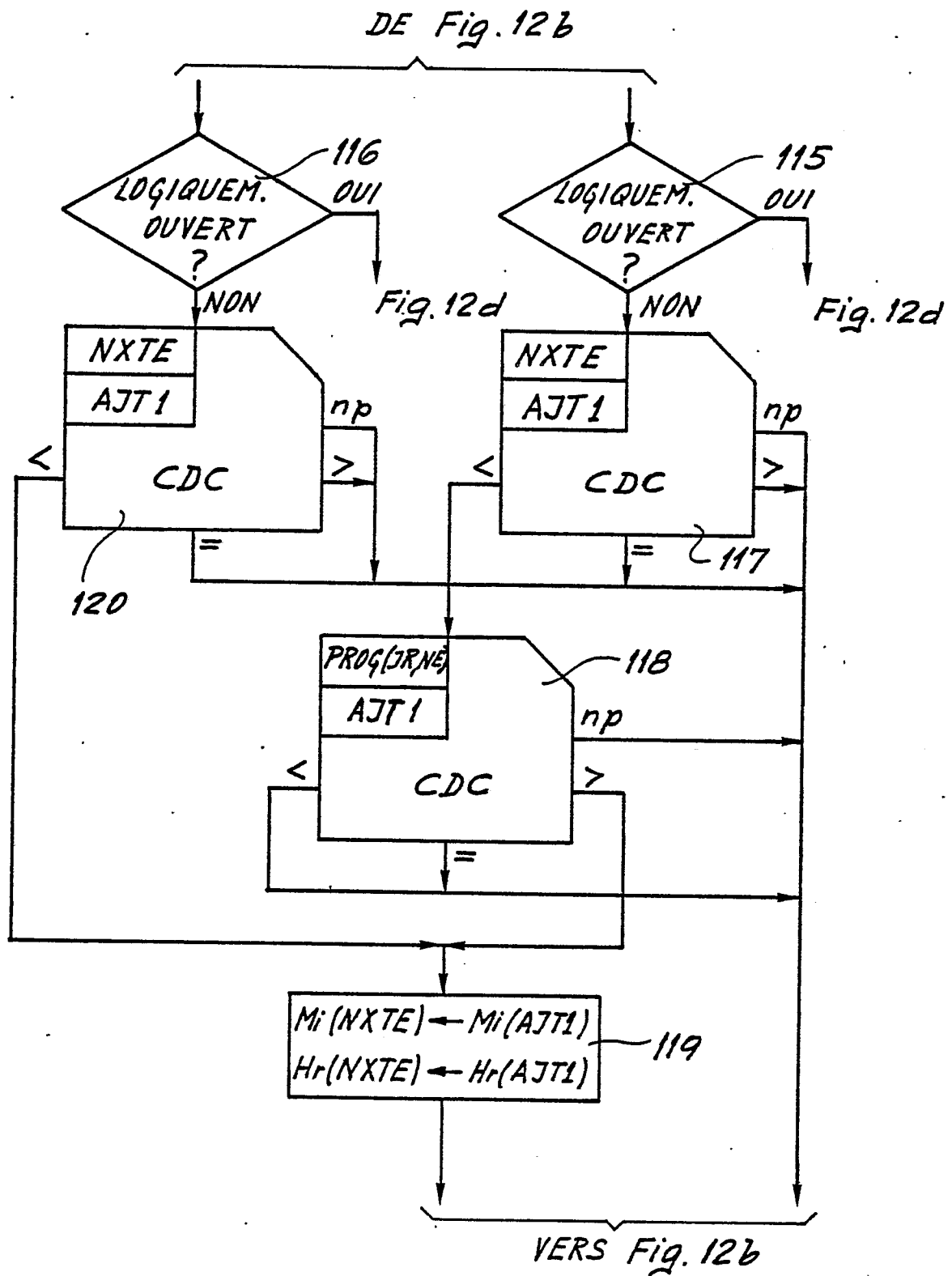


Fig. 12c

DE Fig. 12c

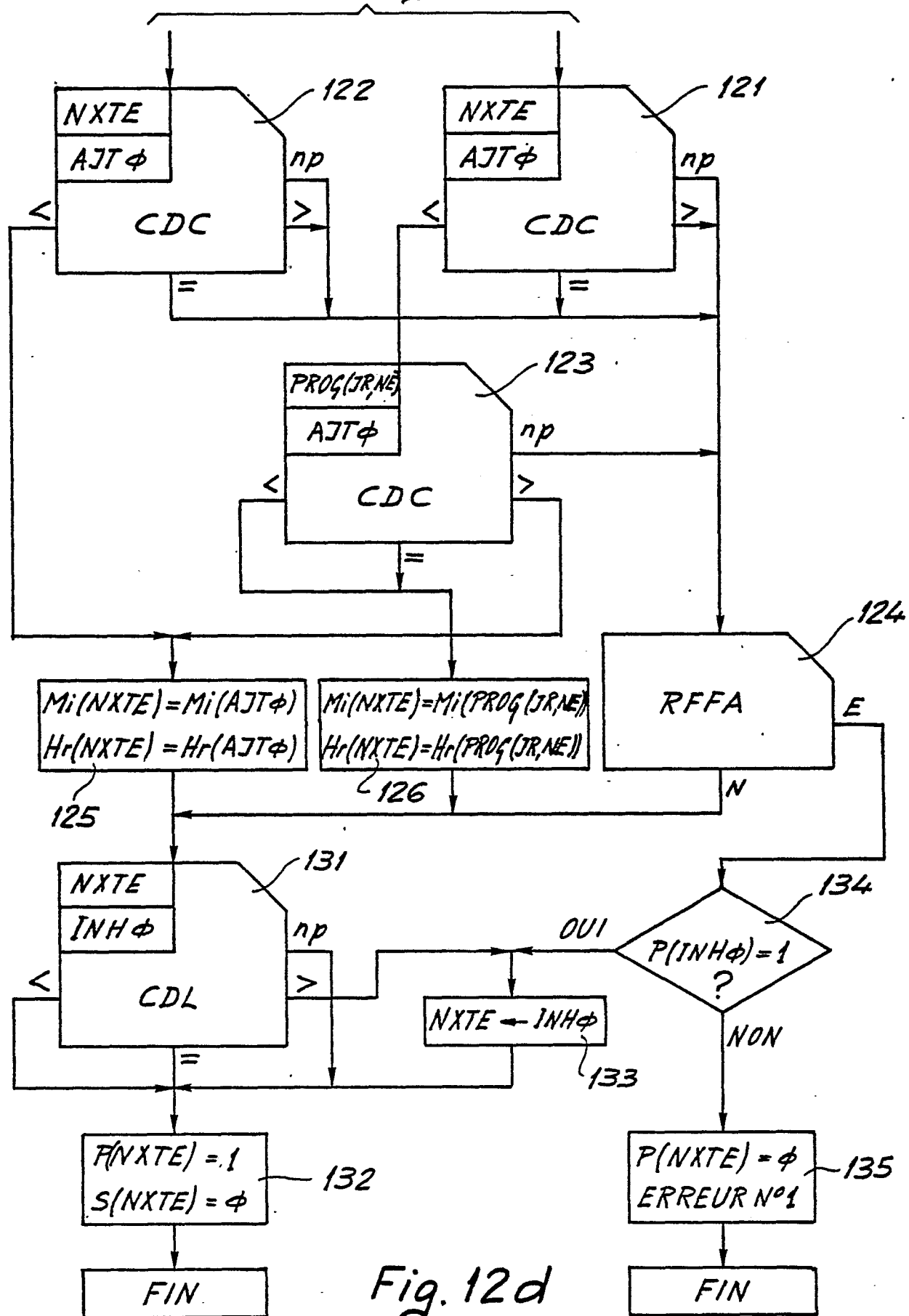


Fig. 12d

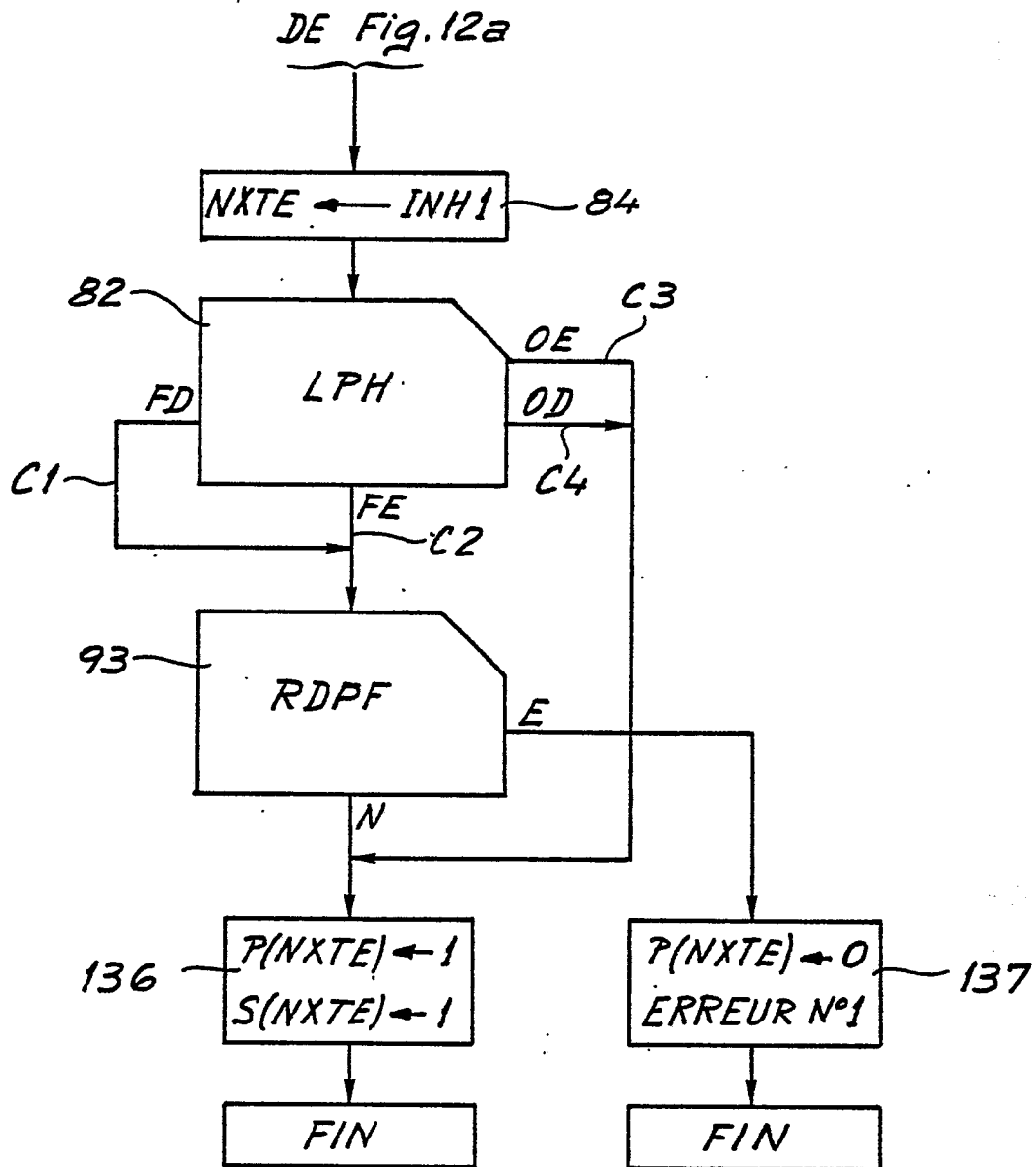


Fig. 12e

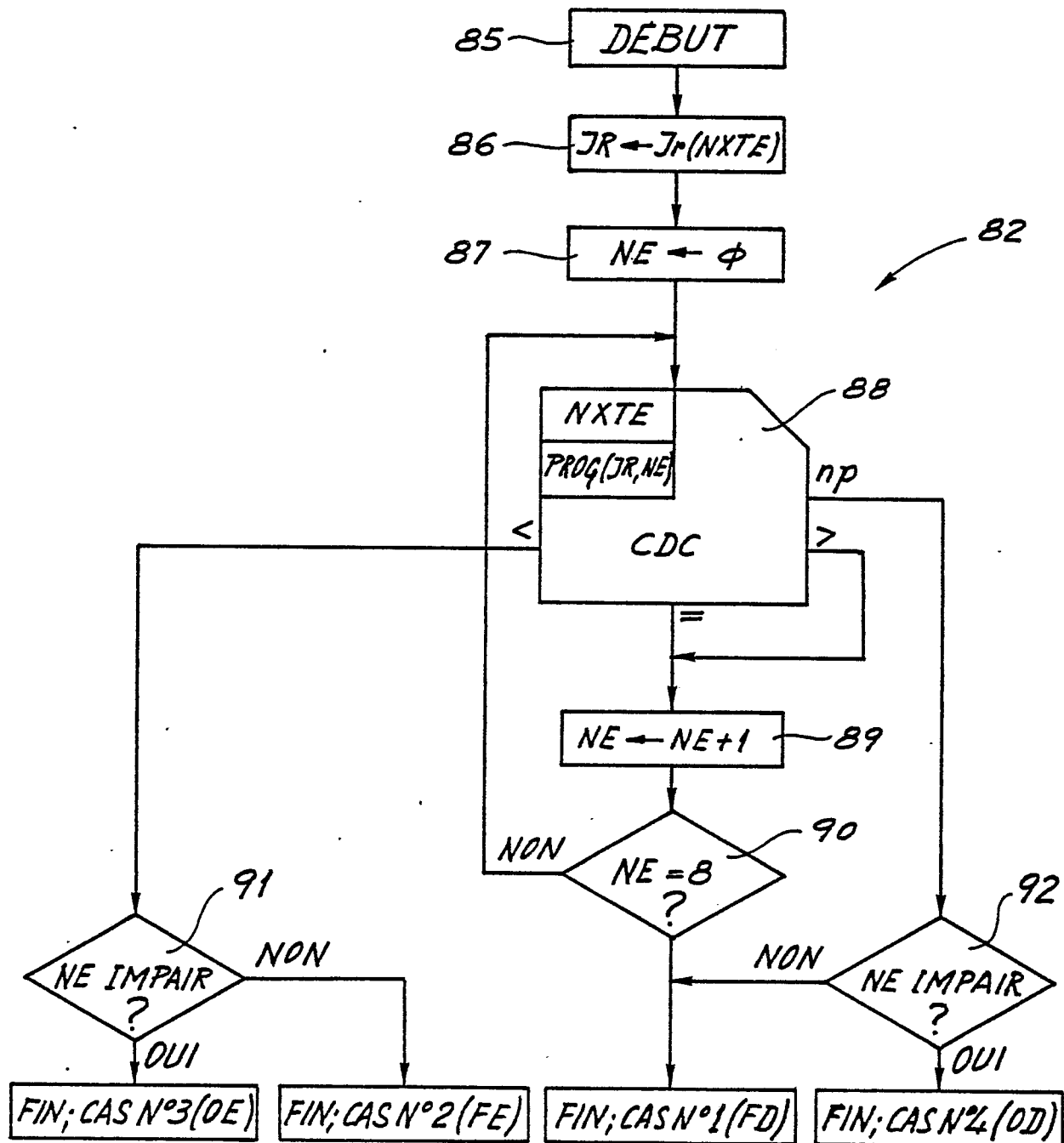


Fig. 13

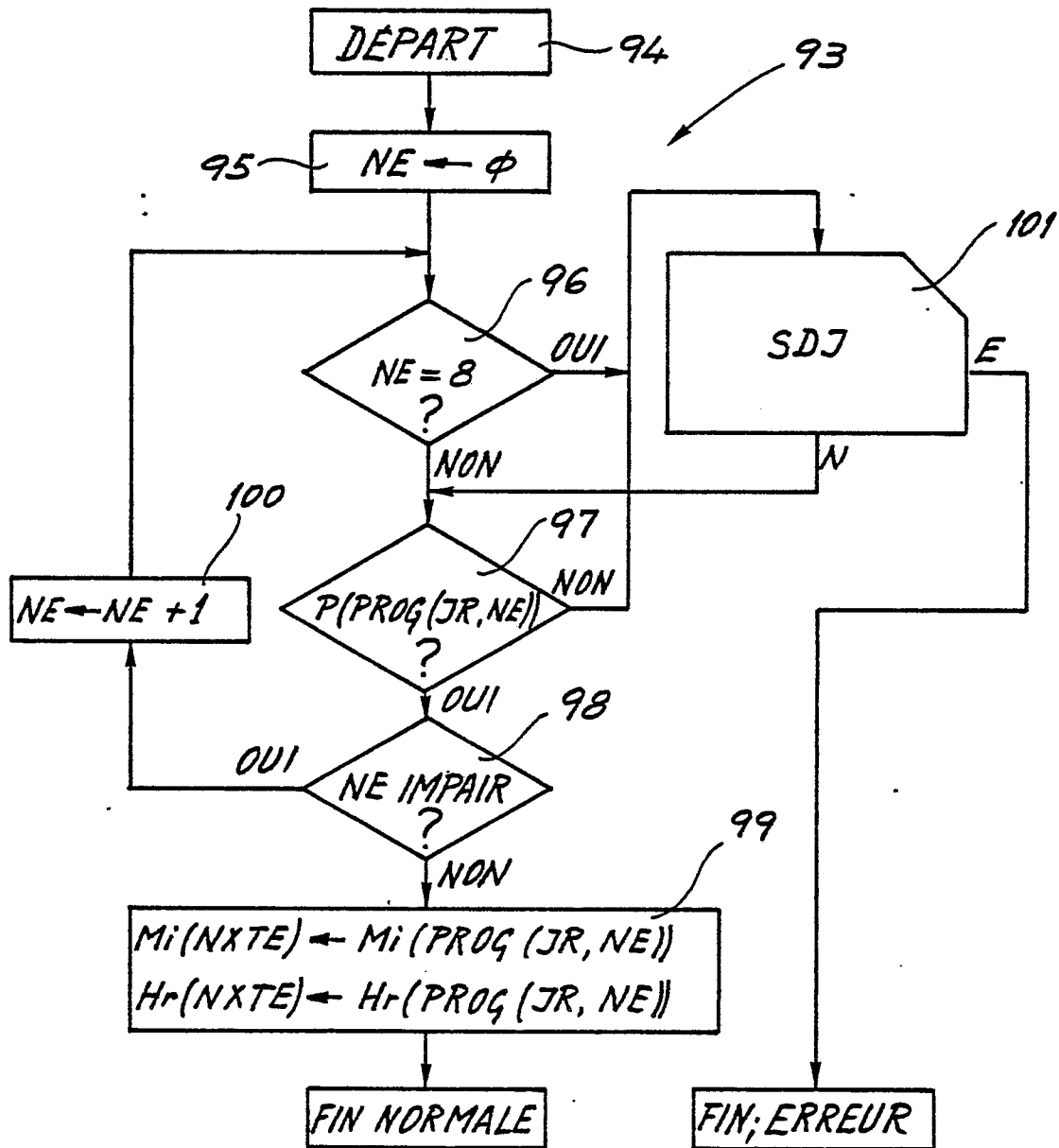


Fig. 14

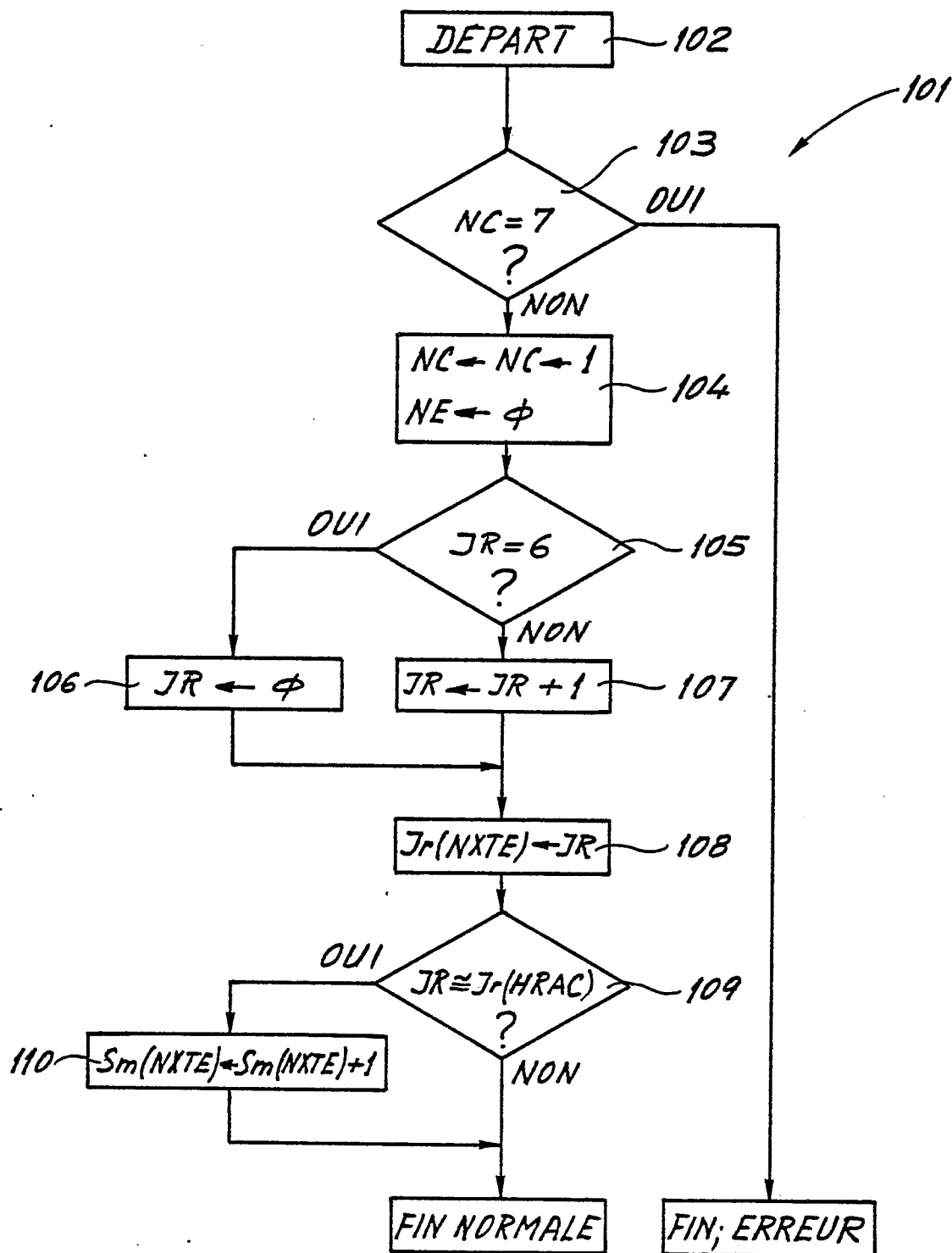


Fig. 15

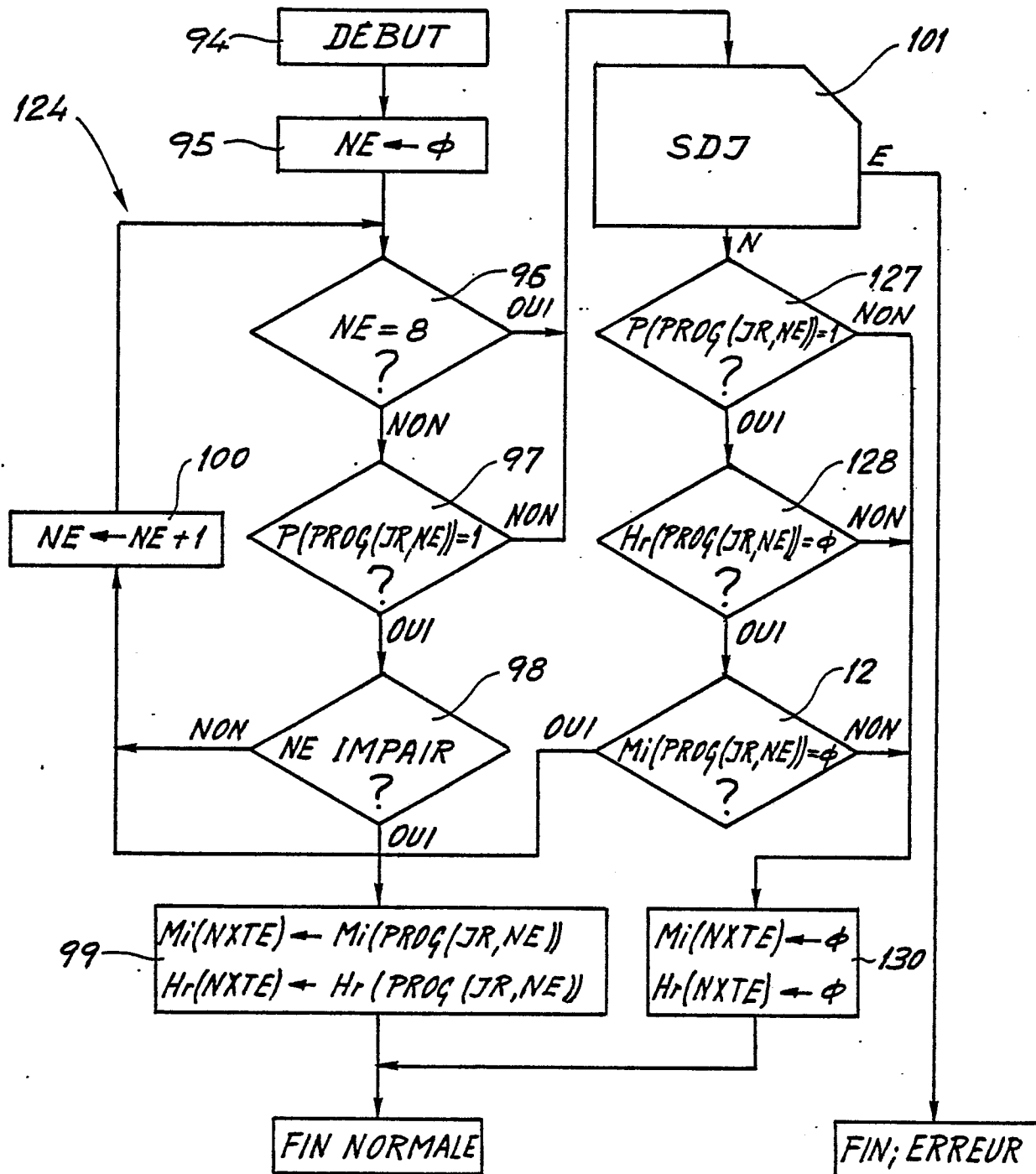


Fig. 16

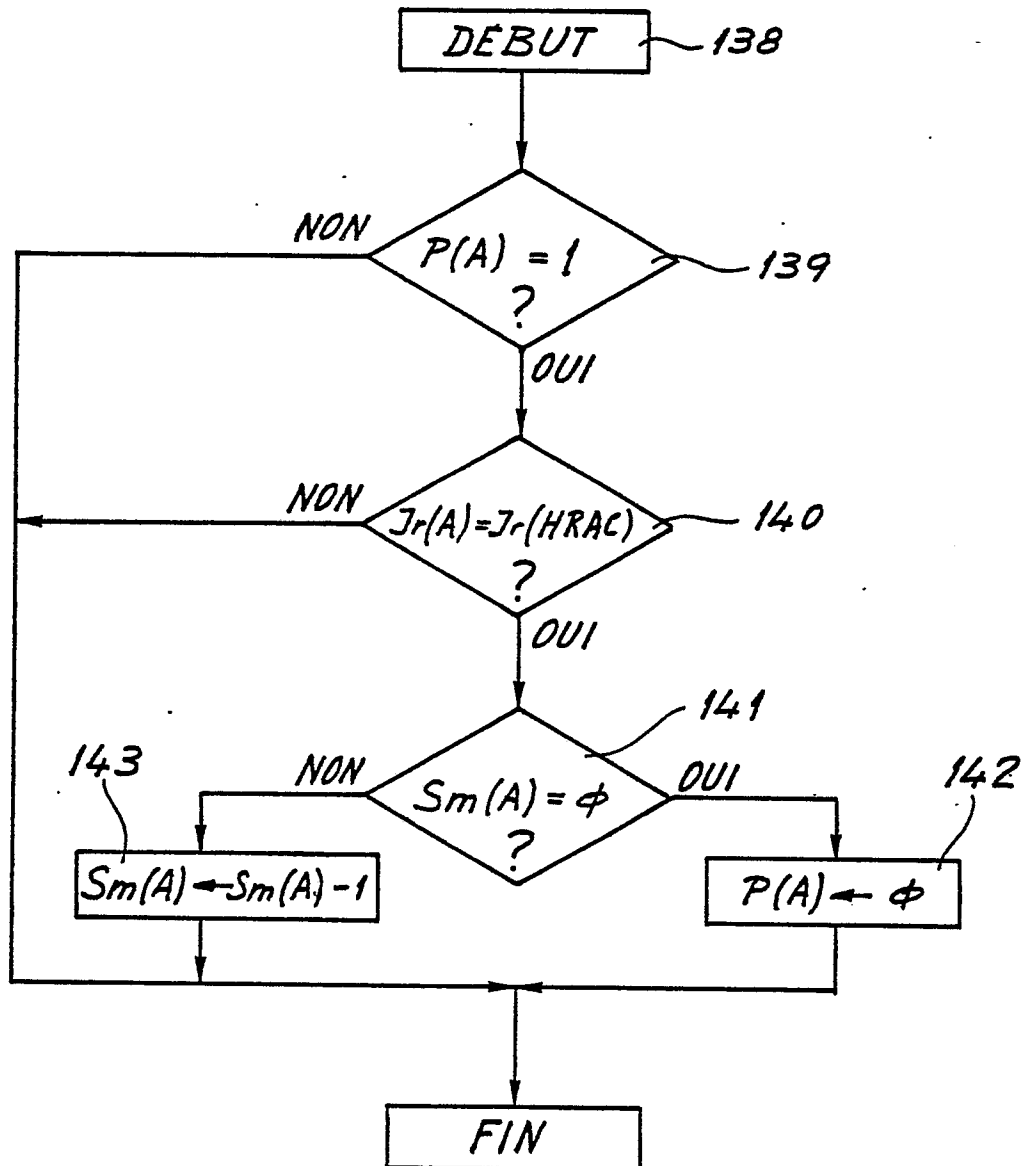


Fig. 17

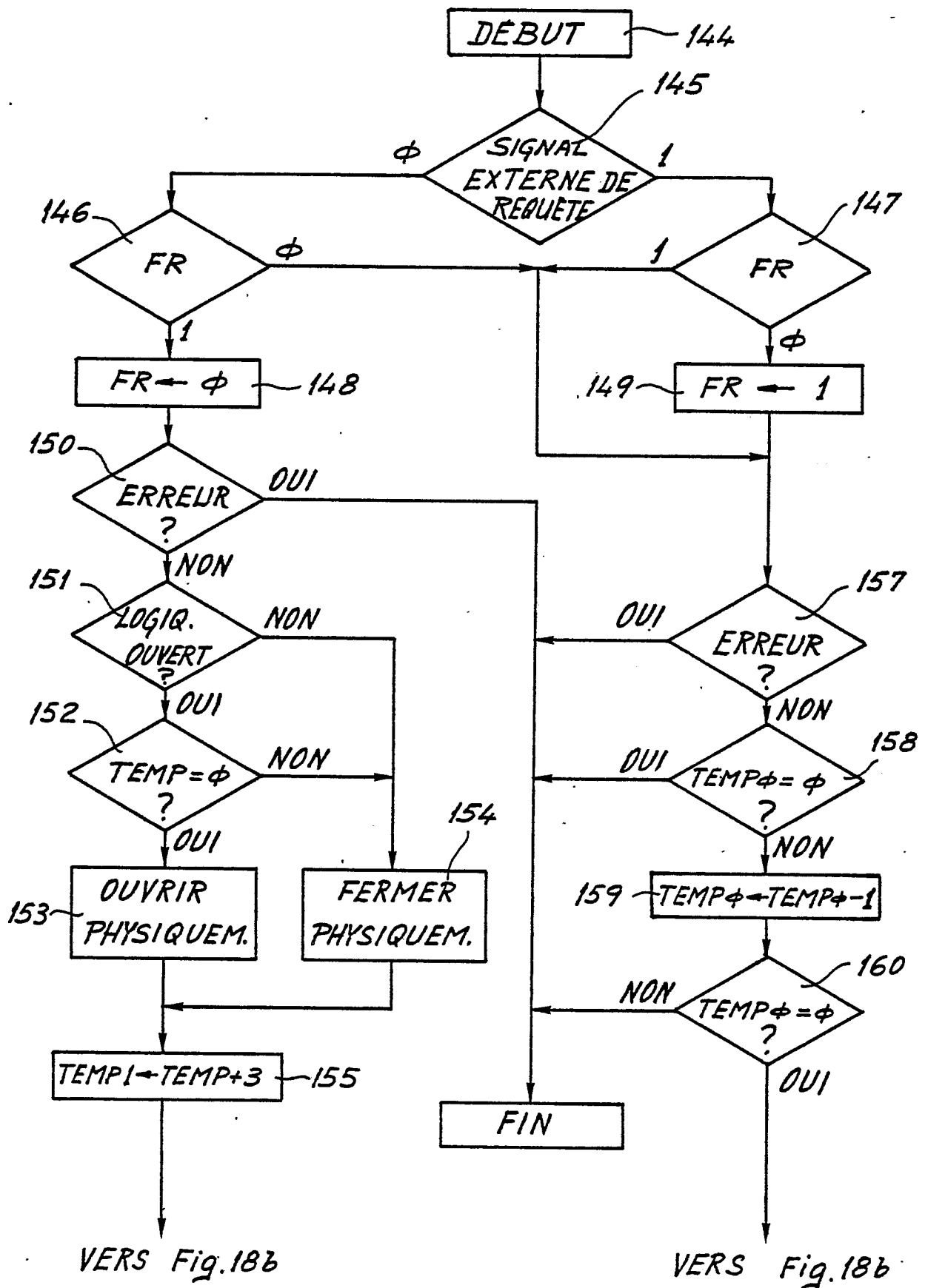


Fig. 18a

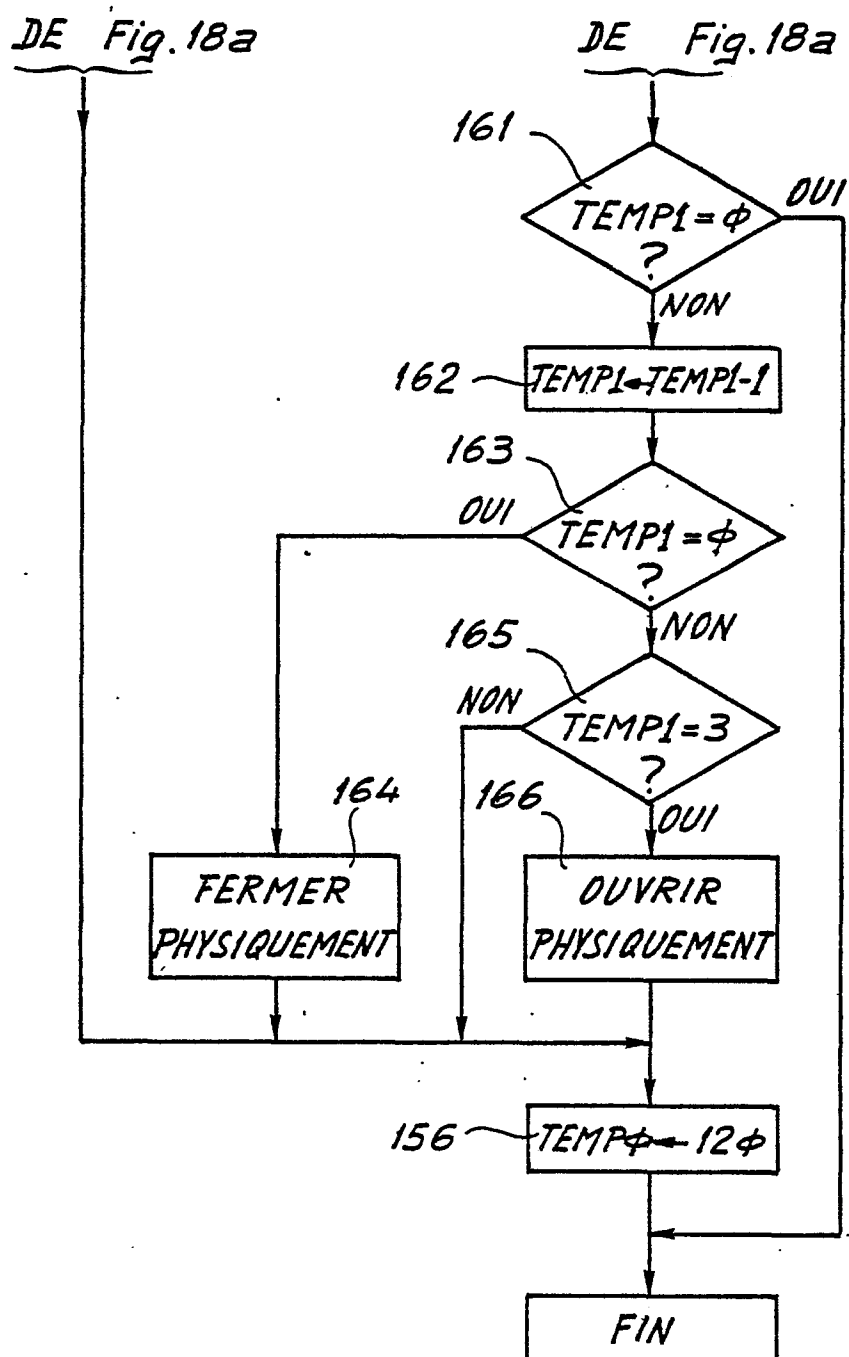


Fig. 18b



DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int. Cl. 4)
Y	US-A-3 881 171 (MOORMAN, WAGNER) * Figure 2; colonne 8, ligne 45 - colonne 9, ligne 25 *	1	E 05 B 49/00
A	---	3,7	
Y	US-A-4 293 915 (CARPENTER et autres) * Figures 1,2; colonne 4, ligne 4 - colonne 12, ligne 15 *	1	
A	---	2,3,7	
Y	ELEKTOR, vol. 9, no. 4, avril 1983, pages 442-450, Canterbury, Kent, GB; "7-day timer/controller", * En entier *	1	DOMAINES TECHNIQUES RECHERCHES (Int. Cl. 4)
A	IDEM	2,3,4	E 05 B G 06 F G 04 G
A	---	1,10	
A	US-A-4 472 789 (SIBLEY) -----	1,2	
Le présent rapport de recherche a été établi pour toutes les revendications			
Lieu de la recherche LA HAYE		Date d'achèvement de la recherche 25-06-1986	Examineur HERBELET J.C.
CATEGORIE DES DOCUMENTS CITES			
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	