

A1

**DEMANDE
DE BREVET D'INVENTION**

(21)

N° 79 31453

(54)

Opérateur autonome de gestion de files d'attente ou de mémoire.

(51)

Classification internationale (Int. Cl. ³). G 11 C 9/04; G 06 F 9/06; H 04 H 1/00.

(22)

Date de dépôt..... 21 décembre 1979.

(33) (32) (31)

Priorité revendiquée :

(41)

Date de la mise à la disposition du
public de la demande..... B.O.P.I. — « Listes » n° 26 du 26-6-1981.

(71)

Déposant : Etablissement public de diffusion dit : TELEDIFFUSION DE FRANCE, ERABIT
Armand, PUT Patrick et RENOULIN Roger, résidant en France.

(72)

Invention de : Armand Erabit, Patrick Put et Roger Renoulin.

(73)

Titulaire : *Idem* (71)

(74)

Mandataire : Brevatome,
25, rue de Ponthieu, 75008 Paris.

La présente invention a pour objet un opérateur autonome de gestion de files d'attente ou de mémoire. Elle trouve une application notamment dans les systèmes de transmissions de données.

5 Le réseau de transmission de données "TRANSPAC" est un réseau public fonctionnant en commutation par paquets. Les réseaux de ce genre comprennent des artères spécialisées véhiculant les données et des dispositifs de commutation. Ces systèmes mettent en relation des
10 calculateurs numériques avec des terminaux informatiques ou des calculateurs entre eux.

La présente invention concerne un moyen de gestion appelé "opérateur" qui peut prendre place dans de tels systèmes où sont constituées des files d'attente
15 de paquets de données et où sont gérées des mémoires de stockage. La gestion de telles files ou de telles mémoires fait appel en général à des logiciels d'allocation et de restitution de données ou de zones de mémoire. Ces logiciels constituent une charge importante
20 dans la tenue à jour des mémoires-tampon disponibles, dans leur attribution à des paquets de données déterminés, dans la création de liens entre pages, etc.... Cette charge est d'autant plus lourde dans les systèmes à commutation par paquets que le problème de la gestion
25 de mémoire y revêt une grande importance.

Pour tout ce qui concerne les réseaux de transmission de données fonctionnant en commutation par paquets et les moyens informatiques qui leur sont associés, on pourra se reporter à l'ouvrage de C. MACCHI
30 et J.F. GUILBERT intitulé "Téléinformatique" et publié par DUNOD, Paris 1979.

Les interventions de l'opérateur inséré dans de tels systèmes sont ignorées des utilisateurs du réseau et n'ont donc aucune implication dans les pro-
35 grammes, les procédures ou les logiciels d'acheminement

des données. L'opérateur de l'invention intervient dans les commutateurs du réseau comme un organe transparent dont le temps de transit est négligeable. Les fonctions de cet opérateur sont :

- 5 - d'améliorer la souplesse des pointeurs de files en ne tenant pas compte de leur gestion qui fait partie intégrante de telles files d'attente ;
- d'optimiser à tout instant l'allocation des pages mémoires disponibles en gérant ces mémoires comme
- 10 des mémoires virtuelles.

Les paramètres fondamentaux qui caractérisent un commutateur de paquets sont, d'une part, sa puissance de commutation exprimée en nombre de paquets par seconde et, d'autre part, le temps de transit des données à

15 travers lui. L'optimisation d'un tel commutateur doit donc porter à la fois sur sa capacité de mémoire-tampon et sur sa rapidité de transfert des paquets d'une source à un destinataire.

En raison des caractéristiques de régulation

20 de flux et de contrôle d'erreur par répétition éventuelle, les données doivent être mémorisées sous forme de paquets organisés en file d'attente. Il apparaît que, si l'on veut minimiser la capacité de stockage nécessaire dans un commutateur, il faut :

- 25 - une gestion dynamique de la mémoire de stockage pour optimiser son utilisation par découpage de la ressource en pages suffisamment petites de façon à éviter les espaces inoccupés ; ce découpage des tampons en pages implique un chaînage de ces pages en même
- 30 temps qu'une mise en commun entre tous les utilisateurs du lot de pages disponibles ; cette allocation des pages mémoires se traduit par la nécessité de gérer des listes de tampons disponibles de façon aussi efficace que possible par des outils appropriés pour
- 35 ne pas allonger les tâches de stockage des données ;
- une mise en file d'attente des paquets en attente de commutation et l'extraction aussi rapide que pos-

sible des requêtes de commutation dès que sont réunies les conditions de transfert vers le destinataire.

Dans l'art antérieur, deux types de matériels ont été utilisés pour remplir les fonctions indiquées

5 ci-dessus :

- les mémoires à accès direct (en terminologie anglo-saxonne "random access memories" ou RAM) (en technologie MOS ou TTL) gérées par microprocesseur ;
- les systèmes câblés de certains boîtiers spécialisés à intégration à grande échelle (LSI). Il s'agit par exemple des boîtiers 2813 du constructeur "Advanced Micro Devices" désignés par le terme "mémoire FIFO" (pour "First-In-First-Out" caractérisant une procédure du type "premier entré, premier sorti").

15 Dans le premier type de système, la manipulation des cellules mémoire consiste en l'écriture, la lecture, la création de liens entre cellules et la tenue à jour des cellules disponibles et des longueurs de files. Elle est réalisée par programme. A titre
20 d'exemple, les opérations de stockage et d'extraction demandent au moins chacune 15 instructions, soit 30 μ s environ. Si le microprocesseur est utilisé à d'autres fonctions comme la gestion de procédures de réseau, ce délai, ainsi que la complexité du logiciel, sont
25 fortement augmentés.

Dans le second type de système, à chaque file d'attente est attribué un certain nombre de boîtiers FIFO spécifiques suivant le nombre d'éléments binaires des mots à stocker (largeur de file) et la
30 longueur maximale admise pour une file (profondeur). Une logique câblée sert d'interface avec l'utilisateur pour la sélection de la file concernée. Ce système, volumineux et coûteux effectue les opérations de stockage et d'extraction en 1 μ s environ dans l'état actuel
35 de la technique.

L'invention a justement pour objet de remédier à ces inconvénients en proposant un opérateur qui présente les avantages suivants :

- rapidité : par rapport à un système à base de micro-processeurs, le temps de cycle de l'opérateur étant de 160 nanosecondes, une opération d'écriture est réalisée en 1,12 microseconde au plus et une opération de lecture en 0,96 microseconde, au lieu de 30 microsecondes environ pour une telle opération commandée par programme ;
- autonomie : l'invention dispose de sa propre base de temps et il est donc autonome par rapport au reste du système, ce qui simplifie l'exploitation de la machine ;
- simplicité : l'interface avec l'organe utilisateur est simple et asynchrone, ce qui rend facile l'utilisation de l'invention dans des machines très diverses ;
- souplesse : par rapport à un système câblé à base de boîtiers spécifiques, on peut faire varier de 1 à 256 le nombre de files d'attente gérées par un même opérateur, fixer à un multiple quelconque de 4 éléments binaires la largeur des files d'attente et faire varier les seuils de remplissage fournis à l'utilisateur ;
- encombrement : un montage réalisé a demandé 62 boîtiers, au lieu de 128 FIFOS au minimum auxquelles s'ajouterait une importante logique de sélection et d'interfaces avec l'utilisateur dans le cas du montage câblé.

On peut également porter au crédit de l'invention le fait qu'en mettant en commun les ressources occupées par les files d'attente, on n'utilise, à probabilité de débordement égale, qu'un nombre de cellules beaucoup plus faible. Le calcul, effectué dans l'hypothèse de files indépendantes avec les lois de départ et d'arrivée poissonniennes et un taux de charge $P = 0,6$, montre que pour atteindre une probabilité de saturation

de 10^{-11} , il faut réserver 50 cellules par file si elles utilisent des ressources propres, et 282 pour un groupe de 64 partageant une mémoire commune. Le rapport entre les ressources nécessaires est donc de :

5

$$\frac{64 \times 50}{282} = 11$$

Ceci s'explique par la compensation entre les nombreuses files peu remplies ou vides et les quelques files de longueur importante à un instant donné.

L'opérateur de l'invention est apte à la fois à la gestion de files d'attente et à la gestion dynamique de mémoire. Il existe des liens étroits entre ces deux types de gestion. Le mécanisme de gestion dynamique d'espace mémoire par création d'une liste de pages disponibles peut en effet être apparenté à une mise en file d'attente de cellules pouvant être attribuées à des processus concurrents ou pouvant être restitués par ceux-ci.

De façon plus précise, un opérateur d'allocation dynamique d'espace mémoire a pour fonction d'associer à chaque canal d'entrée correspondant à une ligne les mémoires-tampon nécessaires au rangement des paquets reçus par cette ligne. Ces tampons sont constitués de pages chaînées entre elles par des liens créés par l'opérateur, qui prélève les pages disponibles et les restitue après libération. La liste ainsi associée à chaque canal est donc constituée de pointeurs de pages qui ne sont pas directement connus du processus utilisateur, lequel s'adresse en quelque sorte à une mémoire virtuelle dont il ne connaît pas les adresses physiques correspondant aux espaces attribués aux paquets.

Par ailleurs, un paquet de données est caractérisé par un pointeur d'origine et une description de son type. Les files d'attente de paquets, en instance

de commutation, associées à chaque voie logique active, sont constituées par la liste des paquets attendant l'autorisation de transfert vers le destinataire. Dès que les conditions de transfert sont réunies, on extrait
5 de la file d'attente relative à la voie logique, la cellule contenant le type de paquet et le pointeur d'origine correspondant à la première page du tampon attribué à ce paquet. Le lien ou le chaînage avec les pages suivantes constituant le tampon, est fourni par
10 l'opérateur de gestion dynamique d'espace mémoire.

De tels opérateurs permettent donc bien une optimisation des fonctions de mémorisation dans un commutateur de paquets.

De façon précise, l'invention a pour objet
15 un opérateur autonome de gestion de files d'attente ou de mémoire, cet opérateur étant associé à une mémoire (D) possédant des cellules-mémoires mises en partage entre plusieurs files d'attente de données ou organisée en chaînes de pages attribuables à des processus con-
20 currentes ; cet opérateur est caractérisé en ce qu'il comprend :

A) - un organe fonctionnel de stockage constitué par :

- 25 . une mémoire de chaînage (CH) comprenant des mots de chaînage pour lesdites données ou lesdites pages,
- . trois mémoires d'identification de files d'attente ou de zones mémoires, ces mémoires mémorisant trois paramètres d'identification : longueur, adresse initiale, adresse finale,
- 30 . une mémoire d'adresses comprenant la pile des adresses des zones disponibles dans la mémoire D,
- . un compteur dont le contenu indique la première adresse disponible dans la mémoire d'adresses ;

35 B) - un organe de séquençement apte à commander ledit organe fonctionnel de stockage.

De préférence, cet organe de séquençement est constitué par :

- 5 . un registre de synchronisation qui fonctionne sous l'action de signaux de commande concernant des demandes d'écriture ou de lecture dans (D) dans le cas de gestion de files d'attente, et de fourniture ou de restitution de pages de (D) dans le cas de gestion de mémoire, et des demandes d'initialisation générale,
- 10 . une mémoire de décodage de commande, cette mémoire étant reliée au registre précédent,
- 10 . un registre d'adresses commandé par la mémoire de décodage,
- 15 . une mémoire de microprogrammes commandée par le registre d'adresses, cette mémoire comprenant une pluralité de mots qui sont des microinstructions adressées aux moyens de l'organe fonctionnel de stockage (A),
- 15 . une base de temps délivrant des signaux de commande desdits registres et desdites mémoires.

20 De toute façon, les caractéristiques et avantages de l'invention apparaîtront mieux après la description qui suit, d'exemples de réalisation donnés à titre explicatif et nullement limitatif. Cette description se réfère à des dessins annexés sur lesquels :

- 25 - la figure 1 représente schématiquement l'organisation de la partie fonctionnelle de l'opérateur de l'invention dans l'application à la gestion de files d'attente ;
- la figure 2 illustre le processus d'écriture d'un mot dans une file d'attente ;
- 30 - la figure 3 illustre le processus de lecture d'un mot dans une file d'attente ;
- la figure 4 représente la structure de l'organe de séquençement de l'opérateur de l'invention ;
- la figure 5 représente la structure de l'organe de stockage de l'opérateur de l'invention ;
- 35 - la figure 6 illustre trois types d'opérations, respectivement d'initialisation, d'écriture et de lecture ;

- les figures 7a et 7b illustrent deux exemples d'insertion de l'opérateur de l'invention dans des matériels informatiques ;

5 - la figure 8 illustre le principe de l'opérateur dans l'application à la gestion de mémoire ;

 - la figure 9 représente la structure de l'organe de stockage adapté à la gestion de mémoire ;

10 - la figure 10 représente les chemins de données correspondant à une opération de lecture de page.

 L'opérateur de gestion de files d'attente de l'invention effectue le stockage et l'extraction de mots de données dans plusieurs files d'attente distinctes faisant partie d'un organe de téléinformatique.
15 Les mots sont extraits d'une file d'attente de numéro N dans l'ordre où ils ont été stockés, suivant le principe premier entré, premier sorti. Chaque mot stocké occupe une cellule de mémoire vive à accès direct (RAM). On appelle longueur d'une file (LF), le nombre de mots
20 stockés dans cette file.

 L'opérateur est sous le contrôle d'un organe téléinformatique utilisateur de nature quelconque. Cet organe lui fournit les mots de données à stocker et en reçoit les mots de données extraits. L'utilisateur
25 commande l'opérateur en lui fournissant le numéro N de la file d'attente en question et en lui précisant l'opération à exécuter. Ces opérations sont de trois types :

- 1) - Ecriture d'un mot M dans une file :
- 30 Une cellule mémoire disponible est chargée par le mot M, puis placée à la queue de la file. L'opérateur avertit l'utilisateur lorsque LF dépasse les seuils de longueur des files, et lorsque la dernière cellule disponible est occupée.
- 35 2) - Lecture dans une file :
- Le plus ancien mot stocké est fourni à l'utilisateur et la cellule mémoire qui le contenait devient

disponible. Si la file est vide, c'est-à-dire si tous les mots stockés ont été extraits, l'opérateur avertit l'utilisateur.

3) - Initialisation générale :

5 Toutes les files d'attente sont vidées et toutes les cellules de la mémoire deviennent disponibles.

A la suite d'une commande, l'opérateur avertit l'utilisateur de ce que la réponse est disponible, et cela sous forme d'avertissements et éventuellement de mot de données extrait. Il lui indique également
10 lorsqu'il est prêt à accepter une nouvelle commande.

L'interface de commande entre l'opérateur de l'invention et l'organe utilisateur est asynchrone et utilise les signaux suivants :

15 - de l'utilisateur vers l'opérateur :

N : numéro de la file concernée,
C : type de commande,
DI : mot de données à écrire (à stocker).

- de l'opérateur vers l'utilisateur :

20 VIDE : file N vide,
PLEIN : la dernière cellule disponible vient d'être chargée,
LF : la longueur de la file N dépasse $L_1 = 16$,
 $L_2 = 32$, $L_3 = 64$, etc...,
25 DO : mot de données lu (extrait),
VS : validation des signaux ci-dessus,
P : opérateur prêt à recevoir une commande,
DINV : commande invalide.

La longueur des mots de données peut être
30 fixée à n'importe quelle valeur multiple de 4 éléments binaires.

L'opérateur de l'invention apte à effectuer ces opérations comprend essentiellement deux organes :

- l'un fonctionnel, de stockage,
- 35 - l'autre de séquençement.

Ces deux organes sont maintenant décrits en référence aux figures 1 à 6.

L'organisation de la partie fonctionnelle est tout d'abord illustrée par les figures 1, 2 et 3. Sur ces figures, les rectangles LF, AI, AF, D, CH et AD représentent différentes mémoires à accès direct.

5 La mémoire D est une mémoire vive dite "de données". Cette mémoire est partagée entre toutes les files d'attente pour le stockage des mots de données. A chaque cellule de D, d'adresse ad, est associée un élément d'adresse ad dans une autre mémoire vive CH
10 dite "mémoire de chaînage".

A un instant donné, chaque file d'attente, de numéro i comprend autant de cellules que de mots de données stockés, soit un nombre $lf(i)$, lequel peut d'ailleurs être nul. L'élément de chaînage associé
15 à la cellule ad où se trouve le mot de données dn, contient l'adresse ads de la cellule où est stocké le premier mot qui suit dn dans la file, soit $dn + 1$; ads peut prendre n'importe quelle valeur différente de ad.

20 Chaque file est identifiée par trois paramètres, rangés sous le numéro de file i dans trois mémoires vives distinctes :

- le nombre de mots stockés qui est conservé dans la mémoire "longueur des files" (LF),
- 25 - l'adresse du premier mot stocké (le plus ancien) qui est conservée dans la mémoire "adresse initiale" (AI),
- l'adresse du dernier mot stocké (le plus récent) qui est conservé dans la mémoire "adresse finale" (AF).

30 Le nombre total des cellules attribuées est conservé dans un compteur CAD adressant la mémoire AD. Les adresses des cellules disponibles sont conservées dans une zone de la mémoire vive AD dite "pile des adresses disponibles" c'est la zone d'adresses supérieures ou égales à la valeur contenue dans le comp-
35 teur CAD.

L'exécution des opérations d'écriture et de lecture est illustrée par les figures 2 et 3 :

1) - Ecriture d'un mot de données M dans la file de rang i (stockage) :

- 5 Il est supposé sur la figure 2 et pour simplifier à l'extrême, qu'à l'instant précédant l'écriture, la file de rang i est constituée de quatre mots, d'adresse ad1 à ad4. La mémoire de longueur de file LF contient donc, à l'adresse i, le nombre 4 ;
- 10 la mémoire d'adresse initiale AI contient, à l'adresse i, un contenu ad1 et la mémoire d'adresse finale AF un contenu ad4. La mémoire de données D contient, aux adresses ad1 à ad4 les quatre mots d1, d2, d3 et d4.
- 15 La mémoire de chaînage CH contient, à l'adresse ad1', l'adresse ad2' du second mot de la file stockée dans D, à l'adresse ad2' l'adresse ad3' du troisième mot de la file et à l'adresse ad3' l'adresse du dernier mot de la file. La pile d'adresses disponibles dans AD commence à ad5.
- 20 Le déroulement de l'opération d'écriture d'un mot est alors le suivant :
- A l'adresse pointée par CAD dans la pile des adresses disponibles AD, est lue l'adresse d'une cellule disponible de la mémoire de données D. Dans l'exemple illustré, cette adresse est ad5. M est rangé à cette adresse (trajets a sur la figure 2).
- 25 La cellule d'adresse ad (ici ad5) doit être chaînée à la queue de la file. Pour cela, l'adresse ad5, qui est maintenant l'adresse finale, est introduite à l'adresse i dans la mémoire AF et chargée dans la mémoire CH à l'adresse ad4 (trajet b).
- 30 Si la file i avait été vide, ad5 deviendrait l'adresse initiale de la file et serait chargée à l'adresse i dans la mémoire AI.
- 35 La longueur de file, lue dans LF à l'adresse i est augmentée d'une unité (ici $4+1 = 5$) et la

nouvelle valeur est chargée dans la mémoire LF (flèche c).

Le compteur CAD indiquant le nombre de cellules utilisées, est incrémenté d'une unité (flèche d).

S'il n'y a plus de cellules disponibles dans la mémoire de données D, le compteur CAD atteint son maximum CAD_{max} et l'opération d'écriture n'est pas réalisable. L'organe utilisateur en est averti.

10 2) - Lecture d'un mot de données dans la file i :

La situation initiale est supposée être celle qui est obtenue après l'opération d'écriture précédente.

15 L'adresse, dans la mémoire de données D, du mot à fournir à l'utilisateur est celle du mot le premier stocké dans la file. Elle est lue à l'adresse i de la mémoire d'adresse initiale AI (ad1). Le mot de données d1 correspondant est lu et fourni à l'utilisateur (trajet a).

20 La nouvelle adresse initiale de la file devient celle du mot suivant (ad2) contenu dans la mémoire de chaînage CH à la même adresse ad1' que l'élément qui vient d'être lu. La mémoire AI est donc chargée à l'adresse i par le contenu de la mémoire CH à l'adresse ad1 (dans le cas illustré ce contenu est ad2) (trajet b).

25 La longueur de la file, lue dans la mémoire LF à l'adresse i est décrétementée d'une unité puis chargée à nouveau dans la mémoire LF (ici $5-1 = 4$).
30 La cellule d'adresse ad1 étant à nouveau disponible dans D, ad1 est placée au sommet de la pile des adresses disponibles dans AD : le compteur CAD est décrétementée d'une unité et ad1 est chargée dans la mémoire AD à l'adresse pointée par CAD (trajet c).

35 Si la file i avait été vide (la longueur de la file lue à l'adresse i de la mémoire LF aurait

été nulle), l'opération de lecture n'aurait pu être exécutée et l'organe utilisateur en aurait été averti.

3) - Initialisation générale :

5 Cette opération, qui n'est pas illustrée, se déroule de la manière suivante : toutes les files d'attente sont vidées par chargement de 0 à toutes les adresses de la mémoire de longueur de files LF. Toutes les cellules sont considérées comme disponibles
10 et placées dans la pile des adresses disponibles AD. Pour cela, chaque mot de cette mémoire est chargé par son adresse. L'organe utilisateur est prévenu de la fin de l'initialisation.

Après cette description de la procédure utilisée par l'opérateur, les moyens mis en oeuvre vont
15 être décrits plus en détail. Comme indiqué plus haut, l'opérateur de l'invention comprend essentiellement deux organes, l'un de stockage des données et l'autre de séquençement des opérations. Sur les figures suivantes, l'opérateur de l'invention portera la référence
20 globale 400, l'organe de séquençement la référence 500 et l'organe de stockage la référence 600 ; les moyens composant ces deux organes seront repérés par des références numériques appartenant aux centaines
25 correspondantes.

La figure 4 tout d'abord, représente la structure de l'organe de séquençement. Cet organe est piloté par une base de temps 550 qui délivre deux signaux d'horloge h_1 et h_2 de rapport cyclique 1/4. Le signal
30 h_1 est utilisé pour la synthèse des signaux de commande de la partie fonctionnelle et h_2 pour des actions sur le séquenceur. Ces deux signaux ont même fréquence et sont décalés l'un par rapport à l'autre d'une demi-période, ceci afin d'éviter des aléas résultant de
35 la modification des entrées d'un élément au moment où l'on utilise des sorties. Le temps qui sépare deux

fronts montants consécutifs de h_2 est appelé cycle.
La fonction de l'organe de séquençement est de délivrer des signaux de commande aux éléments qui composent l'organe fonctionnel. Pour cela il comprend un registre de synchronisation 501 qui fonctionne sous l'action des signaux suivants :

- I_0 : premier élément binaire d'une microinstruction (voir plus loin),
- h_1 : signal de synchronisation fourni par la base de temps 550,
- DEC : demande d'écriture (émanant par exemple d'un organe de gestion de protocole comme l'organe 210 qui apparaîtra sur la figure 7b),
- DLE : demande de lecture (émanant par exemple d'un coupleur comme le coupleur 220 qui apparaîtra sur la figure 7b),
- INIT : initialisation générale (émanant par exemple de l'organe de gestion de protocole).

L'organe de séquençement comprend aussi une mémoire 502 qui reçoit du registre 501 des signaux d'entrée en phase avec l'horloge locale. Cette mémoire 502 est une mémoire morte de décodage des commandes qui, à partir des signaux DEC, DLE, INIT, élabore cinq signaux respectivement P , A_4 , A_3 , DINV et DEM. La signification de ces signaux est la suivante :

- DEM : indique qu'il y avait une demande de traitement en cours, déjà engagée, ce qui permet d'éviter un second traitement de la même commande ;
- DINV : traduit une demande invalide (combinaison erronée d'éléments binaires dans DEC, DLE, INIT) ;
- A_3 et A_4 sont les éléments de poids fort de l'adresse de programme ;
- P : indique que l'opérateur est prêt à exécuter une nouvelle commande.

Le circuit représenté comprend encore une bascule 503, synchronisée par le signal de base de

temps h_2 et commandée par le signal P. Cette bascule délivre un signal "Prêt" (par exemple à l'organe 210 de la figure 7b).

L'organe de séquençement comprend encore

- 5 un registre d'adresser de mémoire de programme 504 et une mémoire de microprogramme 505. La prise en compte d'une commande émanant de l'utilisateur et l'exécution de l'opération demandée exigent plusieurs cycles (jusqu'à 6). A chaque cycle un mot de microprogramme est lu, 10 dont l'adresse, dans la mémoire 505, est contenue dans le registre d'adresse de la mémoire de programme 504. Ce registre progresse sous l'action de h_2 . Il peut être chargé pour une adresse de saut émise par 502 où il progresse élément binaire par élément binaire.

- 15 Le contenu d'un mot de la mémoire de microprogramme 505 est appelé microinstruction de séquençement. Cette mémoire contient les signaux binaires de commande de la partie fonctionnelle et les signaux binaires de commande du séquenceur, qui déterminent l'adresse 20 de la prochaine microinstruction à exécuter. Les quatre commandes possibles sont :

- progression à l'adresse suivante (incrément d'une unité du contenu du registre 504),
- saut inconditionnel à une autre adresse (chargement 25 de 504),
- saut si une condition binaire A est réalisée, sinon progression,
- saut si une condition binaire B est réalisée, sinon progression,

- 30 A et B étant des fonctions booléennes de l'état des éléments fonctionnels de l'application.

Les commandes du séquenceur sont exécutées sur le front montant du signal d'horloge h_2 .

L'organe de séquençement comprend encore

- 35 un registre de synchronisation des commandes par impulsion 506 alimentée par la mémoire 505. Ces commandes sont

destinées aux dispositifs de l'organe de mémorisation 600. Elles sont actives pendant leur front montant, synchrones avec le signal d'horloge h_1 , et remises à zéro sur le signal d'horloge h_2 . Elles sont destinées
5 à celles des entrées de commande des dispositifs de 600 qui sont actives sur des fronts.

L'organe de séquençement comprend un autre registre 507, de synchronisation des états stables pendant un cycle et synchrones avec le signal d'horloge
10 h_1 et alimenté par la mémoire 505. Ces signaux sont exigés de même que les précédents qui marquent les fronts, par les entrées de commande des dispositifs de mémorisation 600 qui sont actives sur des états. L'organe de séquençement de l'invention fournit ainsi
15 les signaux appropriés à la commande des dispositifs de 600.

Le registre 507 synchronise aussi les signaux utilisés pour la progression des adresses du programme et détermine ainsi les conditions de saut. Ce registre
20 507 alimente encore un multiplexeur 508 qui exploite les conditions de saut dont il a été question ci-dessus.

Enfin, un registre 509 contient les signaux de compte-rendu fournis à l'utilisateur après exécution de la commande. Il s'agit des signaux détaillés plus
25 haut à l'exception de P et de DO. Ce registre est chargé dès que la commande est exécutée et synchrone avec le signal d'horloge h_1 .

La structure du microprogramme de séquençement est la suivante. La séquence principale du microprogramme
30 valide les signaux d'entrée provenant de l'organe utilisateur. Le champ du type de commande C est échantillonné une fois par cycle jusqu'à ce que l'utilisateur affiche une configuration binaire de commande. Celle-ci n'est prise en compte que si C contenait, à l'échantillonnage
35 précédent, la configuration "inactive". Ainsi, la même commande de l'utilisateur ne peut être exécutée deux fois.

Sur détection d'une configuration binaire dans C correspondant à l'une des trois opérations de lecture, d'écriture ou d'initialisation, la sous-séquence correspondante est exécutée. Les trois sous-séquences
5 commandent les actions sur les éléments fonctionnels décrits plus haut et valident les signaux de sortie vers l'utilisateur. Le contrôle revient ensuite à la séquence principale.

En cas de détection dans le champ C d'une
10 configuration invalide, l'utilisateur est prévenu et aucune opération n'est exécutée.

Le schéma de l'organe fonctionnel ou de stockage est représenté sur la figure 5. Cet organe comprend :

- une mémoire 603 comportant des cellules contenant
15 les données stockées dans toutes les files (correspond à D de la figure 1),
- un registre 605 contenant le numéro N de la file concernée par la commande,
- une mémoire d'adresse 620 contenant l'adresse de
20 la plus ancienne cellule remplie pour chaque file (correspond à AI de la figure 1),
- une mémoire d'adresse 601 contenant l'adresse de la plus récente cellule remplie pour chaque file (correspond à AF de la figure 1),
- 25 - une mémoire 602 contenant, pour chaque file, le nombre de cellules qui lui sont attribuées (correspond à LF de la figure 1),
- un compteur 606 (CLF) qui augmente ou diminue d'une unité le nombre contenu dans 602 (LF),
- 30 - une mémoire 604 de chaînage qui, à chaque adresse de cellule associe l'adresse de la cellule remplie immédiatement après (correspond à CH de la figure 1),
- une mémoire 607 contenant les adresses des cellules disponibles de 603, (AD),
- 35 - un compteur 608 qui indique le nombre de cellules de 603 utilisées pour stocker les données (correspond

à CAD sur la figure 1) ; les adresses des cellules disponibles sont rangées dans 607 aux adresses supérieures ou égales au contenu du compteur 608. Ce compteur est dit pointeur de pile de la mémoire 607.

5 Les compteurs 606 et 608 sont synchrones de l'horloge locale et chargeables.

L'organe fonctionnel comprend encore des registres d'aiguillage 609 (RAD), 610 (RCH), 611 (RI₁) et 612 (RI₂), à sortie trois états. Ces registres servent
10 à sélectionner les organes accédant aux mémoires ; Par exemple l'accès à 604 se fait à partir, soit du contenu de 601, soit du contenu de 620 via 610. Les chemins établis sont différents suivant la sous-séquence en cours entre entrées et sorties des éléments fonctionnels. La sortie
15 desservant une entrée est à faible impédance, les autres sorties sont à haute impédance. Les chemins des données sont ainsi établis au début d'une sous-séquence et maintenus pendant tout le déroulement de l'écriture, de la lecture ou de l'initialisation.

20 L'organe de stockage comprend encore :

- un registre 613 contenant les données à écrire. Les paramètres sont fournis par exemple par l'organe de gestion 210 de la figure 7b et chargés dans les registres, sur commande de l'organe de séquençement ;
- 25 - un registre 614 qui contient le mot de données extrait d'une file de 603 après une opération d'écriture ; il est chargé sur commande de l'organe de séquençement ;
- un comparateur 615 produisant un signal LF = 0 lorsque
30 la file traitée est vide.

Tous ces moyens sont commandés par des signaux délivrés par l'organe de séquençement de la figure 4. Ces signaux de commande sont de deux types : signaux de commande par "état" (représentés par une double
35 flèche sur la figure 5) et signaux de commande par "impulsion" (représenté par une simple flèche). Ainsi :

- un signal WE, commande par impulsion , décide de l'écriture en mémoire,
- un signal CLK, commande par impulsion , décide du chargement des registres,
- 5 - un signal OE, commande par "état", valide les sorties.

Chaque microinstruction comprend 16 éléments binaires utiles numérotés I_0 à I_{15} conformément au tableau I ci-joint.

- Le format des instructions de séquençement
10 est donné dans le tableau II ci-joint.

La figure 6 montre les liaisons entre les organes de la figure 5, telles qu'elles sont établies pour réaliser les trois fonctions dont il a été question : initialisation (a), écriture (b) et lecture
15 (c) ; les organes non impliqués dans chacune de ces opérations n'ont pas reçu de connexions, étant entendu que le schéma du montage reste celui de la figure 5.

A titre explicatif, il peut être indiqué que les demandeurs ont réalisé avec succès un opérateur
20 conforme à l'invention qui portait sur deux groupes de 64 files d'attente. Chaque groupe disposait de ressources propres, à savoir :

- une mémoire vive de 256 cellules de 12 éléments binaires comme mémoire de données D, l'adresse d'une
25 cellule étant donc codée sur huit éléments binaires ;
- deux mémoires vives de 256 mots de 8 éléments binaires, mémoire de chaînage CH et pile des adresses disponibles AD.

Ainsi, la saturation d'un des groupes par
30 utilisation de toutes les cellules de la mémoire MD n'avait-elle pas de répercussion sur l'autre.

En outre, des ressources communes aux deux groupes étaient prévues qui comprenaient :

- un séquenceur,
- 35 - un interface avec l'organe utilisateur,
- des mémoires vives contenant les paramètres des files : adresse initiale AI, adresse finale AF, longueur

de file LF. Elles contenaient chacune 256 mots de huit éléments binaires. Les mots d'adresses 0 à 63 décrivaient les files du premier groupe, les mots d'adresse 128 à 255 pouvaient décrire deux groupes supplémentaires.

Les figures 7a et 7b montrent deux types de matériels téléinformatiques dans lesquels est inclus un opérateur selon l'invention tel qu'il vient d'être décrit.

Sur la figure 7a sont représentés plusieurs organes 100, 101, ...105 qui délivrent des données de téléinformatique à un bus temporel 110, lequel alimente un opérateur 400 selon l'invention. Cet opérateur est constitué ainsi qu'il a été indiqué plus haut, d'un organe de séquençement 500 et d'un organe de stockage 600. Ces moyens sont reliés à un organe 470 de gestion selon la procédure X25 utilisée dans la transmission de données par paquets dans un réseau non représenté 480.

La figure 7b illustre une autre configuration de matériel téléinformatique qui comprend un organe de commutation de données 200 suivi d'un organe 210 de gestion de protocole X25 relié à un opérateur 400 conforme à l'invention. Cet opérateur est réuni à un coupleur multiplexé 220 alimentant des modems 230 et 240, lesquels desservent des équipements terminaux de téléinformatique non représentés, appartenant à des usagers. L'opérateur 400 comprend encore un organe de séquençement 500 et un organe de stockage 600.

Le fonctionnement de cet ensemble est le suivant. Recevant les paquets de données de 200, l'organe 210 de gestion du protocole X25 accède à l'organe de séquençement 500 et à l'organe de stockage 600. Des câbles 463 acheminent la commande d'écriture et 464 le compte-rendu d'exécution, tandis qu'un câble 466 achemine la commande de lecture et un câble 465 le compte-rendu d'exécution. De son côté, l'organe 600

est chargé du rangement des trames formées dont les câbles 461 apportent les données et 462 les numéros de ligne. Le coupleur 220 y puise les trames à émettre par des câbles 467 transportant les données et 468 les numéros des lignes desservies par 220.

Le dispositif qui vient d'être décrit dans l'application à la gestion de files d'attente de données peut être aisément adapté à la gestion dynamique de mémoire. C'est ce qui est décrit maintenant en liaison avec les figures 8 à 10.

Dans ce cas, l'opérateur gère une mémoire vive D utilisée comme moyen de stockage par un grand nombre de processus concurrents. Chaque processus utilise, à un instant donné, une zone de mémoire constituée par des mots de D dans un ordre précis. Si l'on imposait à ces zones mémoire d'occuper des adresses consécutives dans D, on arriverait à une mauvaise utilisation de la mémoire : il resterait un grand nombre de zones séparées, de N mots disponibles, et un processus ayant besoin de N+1 mots ne pourrait les obtenir. On utilise donc une technique dite de pagination dans laquelle la mémoire est divisée en pages comprenant chacune T mots consécutifs. T est appelé taille de la page ; il s'agit pour des raisons de commodité, d'une puissance de 2. A chaque page est associé un pointeur vers une autre page, appelé mot de chaînage : il est ainsi possible d'attribuer des pages selon les besoins de chaque processus. Les liens entre celles-ci sont créés par écriture dans des mémoires de chaînage.

La figure 8 représente schématiquement une mémoire D utilisée par trois processus concurrents P1, P2, P3. Les zones numérotées 1, 2, 3 dans D représentent schématiquement des zones mémoire utilisées pour les processus P1, P2, P3 en question et les zones hachurées des zones disponibles. Le bloc rectangulaire CH représente la mémoire de chaînage. A chaque page de la mémoire D est associée, dans la mémoire CH l'adresse de la page suivante attribuée au même processus. Ces

adresses (de 0 à 11) sont notées sur la partie gauche de la mémoire. Par exemple, à la page 0 attribuée au processus P3 est associée, par chaînage, la page 7 attribuée elle aussi au processus P3 immédiatement après la page 0.

L'opérateur de gestion dynamique de mémoire selon l'invention permet la gestion des zones par attribution et restitution de pages. Il donne aux zones une structure de file d'attente : la page restituée est la plus anciennement attribuée au processus considéré (principe premier arrivé, premier sorti). Le mot de chaînage associé à une page P contient donc l'adresse de la page P' attribuée au processus, immédiatement après P.

L'opérateur est sous le contrôle d'un organe utilisateur de nature quelconque, qui reçoit les adresses de pages nouvellement attribuées ou restituées. L'utilisateur commande l'exécution d'une opération parmi 4 :

- fourniture d'une page au processus N :
 - . L'adresse de début de page est fournie à l'utilisateur, en même temps qu'un avertissement si la taille de la zone dépasse les seuils de 16, 32, 64 pages et si la dernière page est attribuée. La page est placée en queue de la file N (zone attribuée au processus N) ;
- restitution d'une page par le processus N :
 - . L'adresse de la plus ancienne page fournie au processus N est transmise à l'utilisateur. Cette page devient disponible pour fourniture à un processus quelconque. Si la zone est vide, l'opérateur avertit l'utilisateur.
- Lecture d'une page située après une page P :
 - . L'adresse P est fournie par l'utilisateur à l'opérateur, qui indique l'adresse de la page P' attribuée après la page P. P' reste attribuée au processus N, dont la zone mémoire n'est pas modifiée.
- Initialisation générale :
 - . Toutes les zones sont vidées et toutes les pages mémoire deviennent disponibles.

A la suite d'une commande, l'opérateur avertit l'organe utilisateur lorsque la réponse (avertissements, page fournie) est disponible puis lorsqu'il est prêt à accepter une nouvelle commande.

5 L'interface entre l'opérateur et l'organe utilisateur est asynchrone et utilise les signaux suivants :

- de l'utilisateur vers l'opérateur :

N : numéro de processus concerné,
10 C : type de commande,
PI : adresse de la page pour laquelle on veut connaître la page suivante.

- de l'opérateur vers l'utilisateur :

VIDE : le processus N a une zone vide,
15 PLEIN : la dernière page disponible vient d'être fournie,
LF : la zone attribuée au processus N dépasse 16, 32, 64 pages,
PO : page fournie ou restituée,
20 PS : adresse de page suivant PI,
VS : validation des signaux ci-dessus,
P : opérateur prêt à recevoir une commande.

Le nombre de pages dans la mémoire de données D, qui fixe la longueur de l'adresse de page, peut
25 être choisi à n'importe quelle valeur multiple de 4 éléments binaires.

La taille de chaque page (nombre de mots, largeur des mots) est inconnue de l'opérateur qui n'exécute aucune opération dans D. L'utilisateur prend seul
30 la décision d'écrire ou de lire tout ou partie des pages dont les adresses lui sont fournies par l'opérateur. Les mots de chaînage sont stockés dans l'opérateur, ce qui évite toute contrainte sur la structure de la mémoire D.

35 La partie fonctionnelle de l'opérateur applicable à la gestion de mémoire est identique à celle

qui a été décrite en liaison avec la figure 5. Cette organisation est illustrée sur la figure 9. On observera cependant une différence liée au fait que l'écriture et la lecture de D sont à la discrétion de l'utilisateur ; l'adresse ad des cellules, au lieu d'être utilisée pour adresser D, est directement fournie à l'utilisateur dans les trois cas de fourniture, de restitution de page et de lecture de page suivante.

L'identification des zones est faite par les mêmes paramètres que pour les files d'attente, à savoir par une adresse initiale (AI, mémoire 620), une adresse finale (AF, mémoire 601) et une longueur de zone (LF, mémoire 602).

Les opérations de fourniture et de restitution de page se déroulent de la même manière que les opérations de lecture et d'écriture d'une file. Il n'y pas d'action sur la mémoire de données. L'opération d'initialisation générale est identique à celle décrite précédemment. Toutefois, la lecture de page suivante est une opération non prévue dans l'opérateur de gestion de files d'attente. L'utilisateur fournit à l'opérateur l'adresse PI d'une page de la mémoire de données. A cette adresse, dans la mémoire de chaînage CH (PI) est lue l'adresse ad' de la page suivante, qui est fournie à l'utilisateur. Les paramètres de la zone du processus N, à qui ont été fournies les pages PI et ad' ne sont pas modifiés (AI, AF et LF).

Les registres d'interface 613 et 614 (DI et DO) représentés sur la figure 5 ne sont plus nécessaires ici, mais il faut en revanche ajouter :

- un registre 650 à 8 éléments binaires qui contient le numéro (PI) de la page fournie par l'utilisateur pour obtenir le numéro de page suivante,
- un registre 652 à 8 éléments binaires chargé par l'opérateur avec le numéro de page suivante (PS),
- un registre 654 à 8 éléments binaires chargé par l'opérateur avec le numéro de la page fournie ou restituée (PO),

- un bus d'adresses 660.

Par ailleurs, la mémoire 603 (D) comprend une entrée de lecture 670, une entrée d'écriture 672, une entrée d'adresse de page 674 et une entrée de déplacement 676. Les données à écrire sont véhiculées
5 par une connexion 680 et les données lues par une connexion 682.

Le code d'ordre est identique à celui de l'application précédente, mis à part la disparition
10 des actions sur D (603), DI (613) et DO (614); le 13ème élément binaire assure le chargement des registres PI (650), PS (652) et PO (654).

Pour ce qui est de l'exécution des commandes, la configuration 1, 1, 0 des éléments INIT, LEC, EC
15 est interprétée comme une commande de lecture de page suivante. La séquence principale cède alors le contrôle à la sous-séquence "lecture de page suivante" qui dure deux instructions.

Les chemins de données établis pour cette
20 opération de lecture de page suivante sont représentés en traits forts sur le schéma de la figure 10.

L'organe séquenceur de l'opérateur de gestion de mémoire est semblable à celui de la figure 4 ; il effectue en outre une quatrième opération de lecture
25 de page suivante, à laquelle sont associées une configuration binaire dans le champ de commande C et une sous-séquence dans la mémoire de microprogramme (505).

Il va de soi que l'application de l'opérateur de l'invention aux commutateurs de paquets n'a été
30 indiquée qu'à titre explicatif et que cet opérateur peut être utilisé dans bien d'autres systèmes, l'allocation dynamique de mémoire intervenant dans tout processus de stockage temporaire d'information et la mise en files d'attente multiples dans tout système de mul-
35 titraitement exigeant la gestion de tâches pour de multiples usagers.

TABLEAU I

N° de l'élément binaire (poids le plus faible : 0)	Action	Convention (D, directe ; I, inverse) (IM, impulsion)
0	Comptage/décomptage de CAD	D IM
1	Décomptage de CLF	D IM
2	Comptage de CLF	D IM
3	Validation de RCH, RAD, RI ₁ , RI ₂ , ST	D IM
	Chargement de CLF	
4	Ecriture dans CH ou AI	D
5	Ecriture dans AI et AD	D
6	Ecriture dans AF et D	D
7	Ecriture dans LF	D
8	Valeur de l'eb dans VS	D
9	RAZ de CLF	D
10	Condition de saut :	00 : saut inconditionnel
11		01 : pas de saut (incrément)
	AO, eb de poids faible de l'adresse de saut	10 : saut si CLF=0
	Choix entre branchement et retour au programme	11 : saut si CAD minimum
	principal	ou maximum
12	Validation de DE	D
13	chargement de CAD	0 : retour au programme
14		principal
15		I
		I

TABLEAU II

Sous-séquence	Numéro d'instruction symbo- décimal lique	Action sur les éléments fonctionnels		Action sur le séquenceur
		par impulsion	par état	
Séquence principale	AO 00	Chargement des registres 605(n° de file) 613(mot à stocker) 501(demande utilisateur)		Saut incondi- tionnel au début de la sous-séquence choisie par l'utili- sateur (501).
	A1 01		Validation du contenu du registre 509	Retour incondi- tionnel en AO
Sous-séquence d'erreur	A2 02	Chargement du registre 509 avec le signal "commande invalide"		Passage en A3
	A3 03		Validation du contenu de 509	Retour incondi- tionnel en AO
	I0 08		Chargement de 0 dans 606, de 255 dans 608	Passage en I1
Sous-séquence d'initialisation	I1 09	Chargement des registres 509, 611 et 612 (adresse de 607, à initialiser)		Passage en I2

TABLEAU II (suite)

Sous-séquence	Numéro d'instruction symbo- décimal lique	Actions sur les éléments fonctionnels		Action sur le séquenceur
		par impulsion	par état	
	12 10	Ecriture à l'adresse à initialiser ad : de 0 dans 602 de ad dans 607		Saut en A1 (fin) si toutes adresses initialisées (cont.608=0) sinon passage en 13.
	13 11	Décrément de 608 ad - 1 ad		

TABLEAU II (suite)

Sous-séquence d'écriture	Numéro d'instruction symbolique	Action sur les éléments fonctionnels		Action sur le séquenceur
		par impulsion	par état	
	E0	Chargement des registres 509, 606 (longueur de file) 609 (adresse ad d'une cellule de D disponible)		Retour en A1 (fin) s'il n'y a aucune cellule disponible (cont. 608 = 255). Sinon passage en E1
	E1	<p>Ecriture de l'adresse ad :</p> <p>- dans 620 si la file était vide</p> <p>- dans 604 adressée par 601 sinon</p>	Validation du contenu de 509	Passage en E2
	E2	<p>Ecriture du mot à stocker dans 603 à l'adresse ad</p> <p>Ecriture de ad dans 601</p> <p>Incrément de 606</p>		Passage en E3
	E3	<p>Ecriture dans 602 de la nouvelle longueur de file (+1)</p> <p>Incrément de 608</p>		retour en A0 inconditionnel (fin)

TABLEAU II (fin)

Sous-séquence	Numéro d'instruction symbolique	Action sur les éléments fonctionnels		Action sur la séquenceur
		par impulsion	par état	
Sous-séquence de lecture	LO	Chargement des registres 509 606 (longueur de file) 610 (adresse de la cellule lue) 614 (donnée extraite)		Retour en A1 (fin) si la file est vide (LF=0), sinon passage en L1
	L1	Décrément de 1 de 606 et 608 (adresse redevenue disponible)	Validation du contenu de 509	Passage en L1
	L3	Ecriture - dans 620 de l'adresse contenue dans 604 - dans 607 de l'adresse redevenue disponible - et dans 602 de la nouvelle longueur de file (-1)		Retour inconditionnel en A0 (fin)

REVENDICATIONS

1. Opérateur autonome de gestion de files d'attente ou de mémoire, associé à une mémoire (D) possédant des cellules-mémoires mises en partage entre plusieurs files d'attente de données ou organisée en chaînes de pages attribuables à des processus concurrents, caractérisé en ce qu'il comprend :
- 5 A) - un organe fonctionnel de stockage (500) constitué par :
- 10 - une mémoire de chaînage (CH) comprenant des mots de chaînage pour lesdites données ou lesdites pages,
 - trois mémoires d'identification de files d'attente ou de zones mémoires, respectivement LF, AI et AF, ces mémoires mémorisant trois paramètres d'identification, longueur, adresse initiale, 15 adresse finale,
 - une mémoire d'adresse (AD) comprenant la pile des adresses des zones disponibles dans la mémoire D,
 - un compteur (CAD) dont le contenu indique la 20 première adresse disponible dans la mémoire d'adresses AD,
- B) - un organe de séquençement (600) apte à commander ledit organe fonctionnel de stockage.
2. Opérateur selon la revendication 1, caracté-
- 25 risé en ce que l'organe de séquençement est constitué par :
- un registre de synchronisation (501), qui fonctionne sous l'action de signaux de commande concernant des demandes d'écriture ou de lecture dans D dans le 30 cas de gestion de files d'attente, et de fourniture ou de restitution de pages de D dans le cas de gestion de mémoire, et des demandes d'initialisation générale,
 - une mémoire (502) de décodage de commande, cette mémoire étant reliée au registre précédent,
 - 35 - un registre d'adresses (504) commandé par la mémoire de décodage (502),

- 5 - une mémoire de microprogrammes (505) commandée
par le registre d'adresses (540), cette mémoire
comprenant une pluralité de mots qui sont des
microinstructions adressées aux moyens de l'organe
fonctionnel de stockage (A),
- une base de temps (550) délivrant des signaux
de commande desdits registres (501, 504) et
desdites mémoires (502, 505).

1 / 10

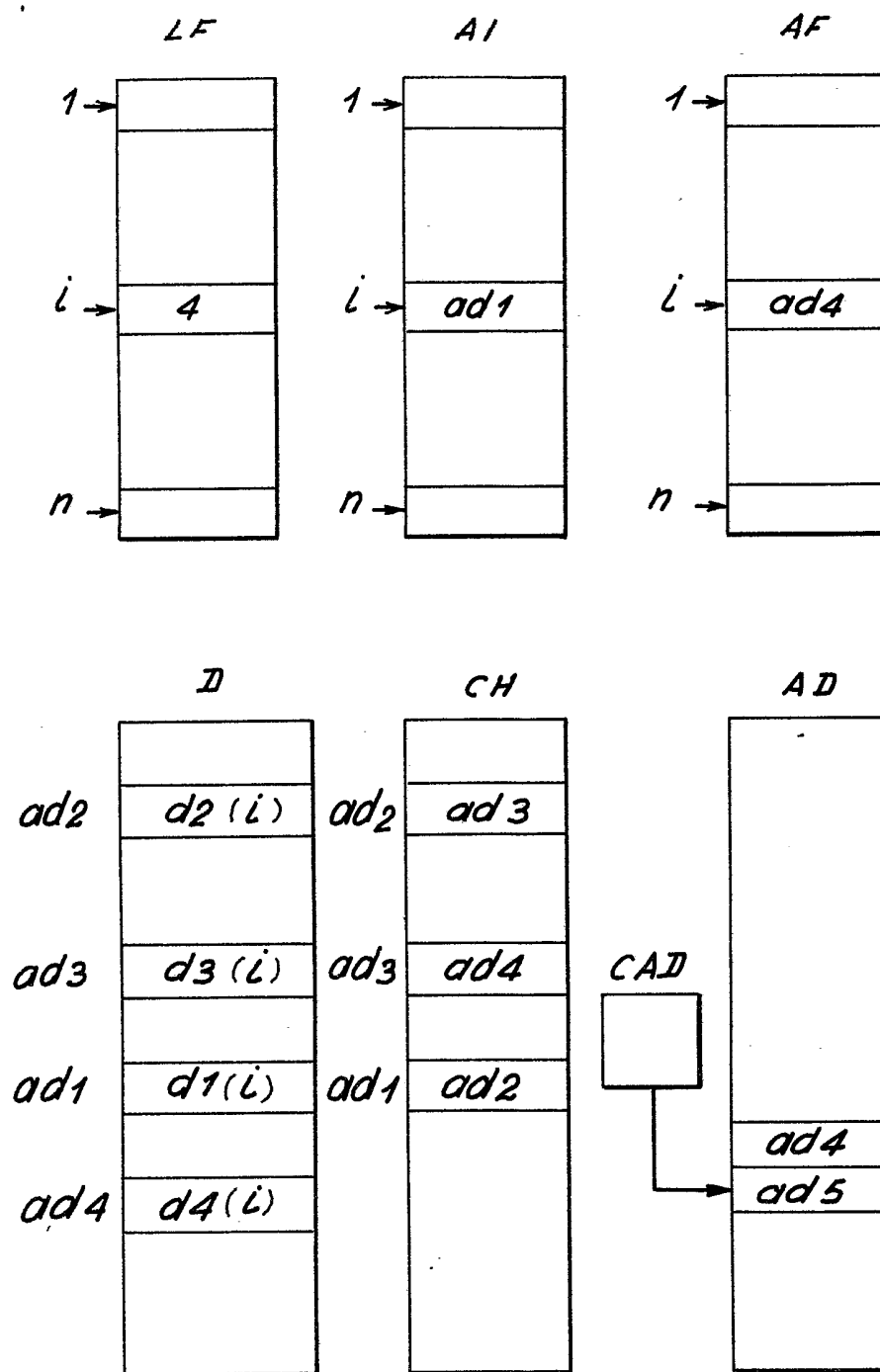


FIG.1

2 / 10

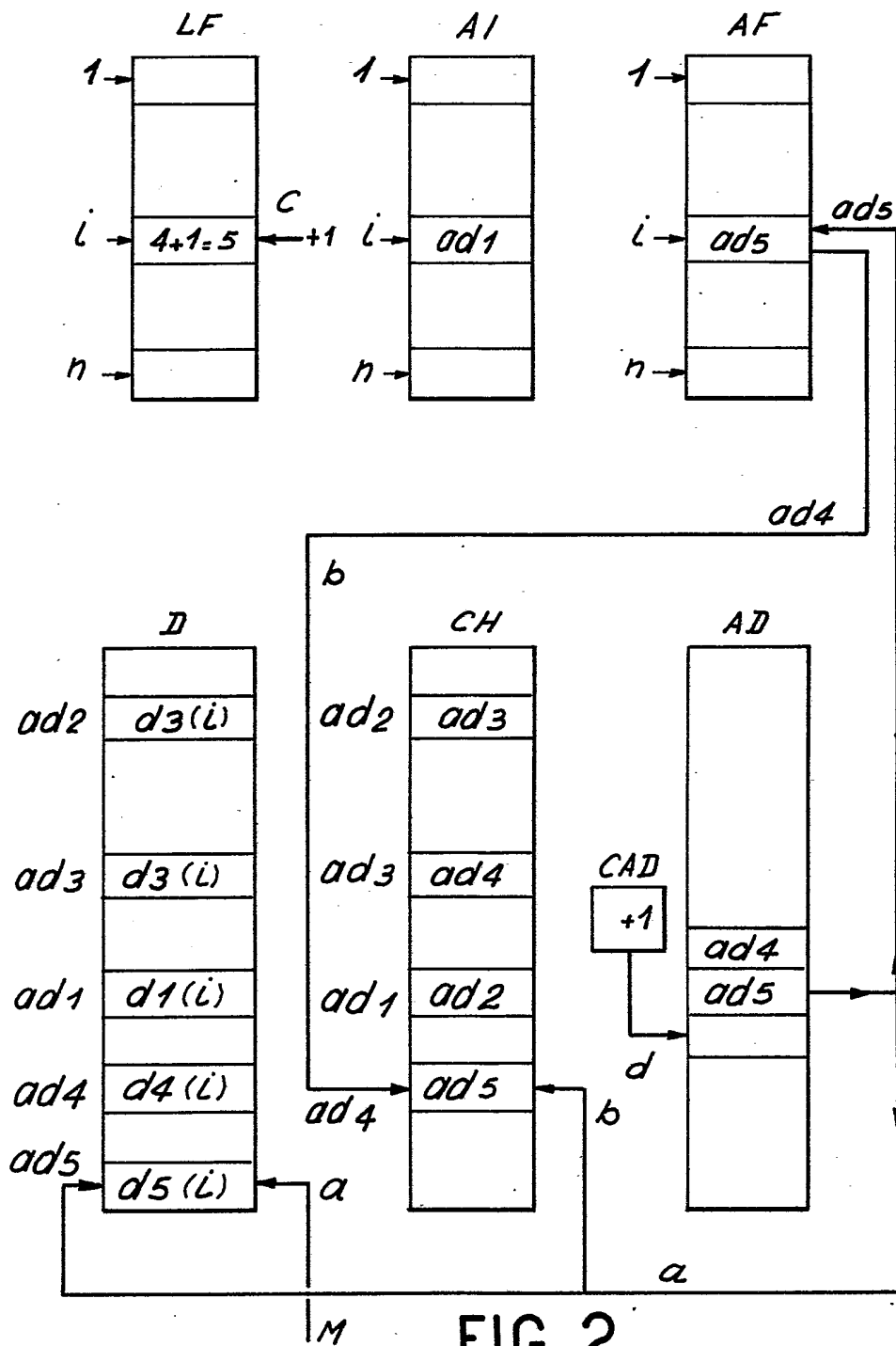
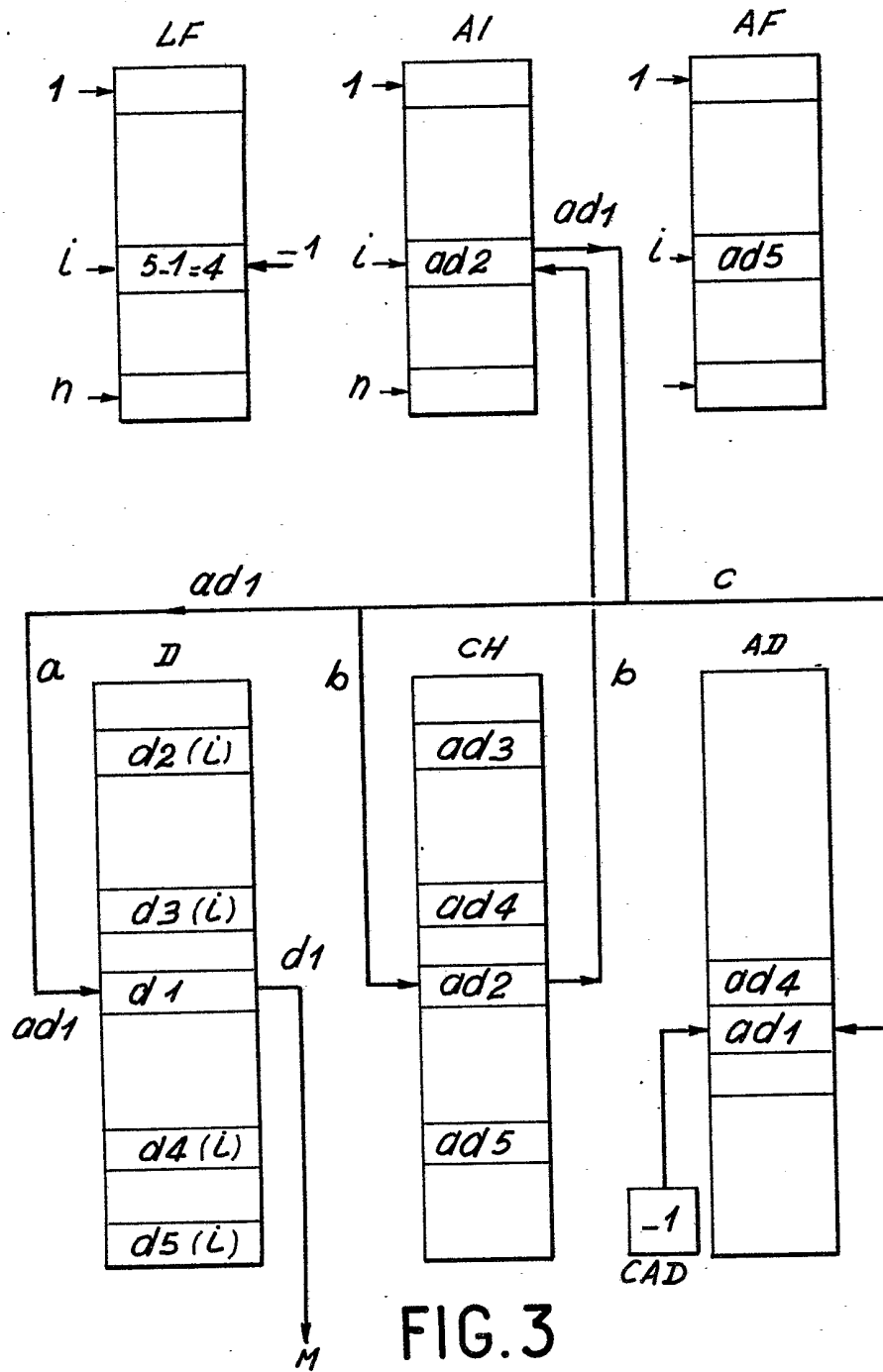


FIG. 2

3 / 10



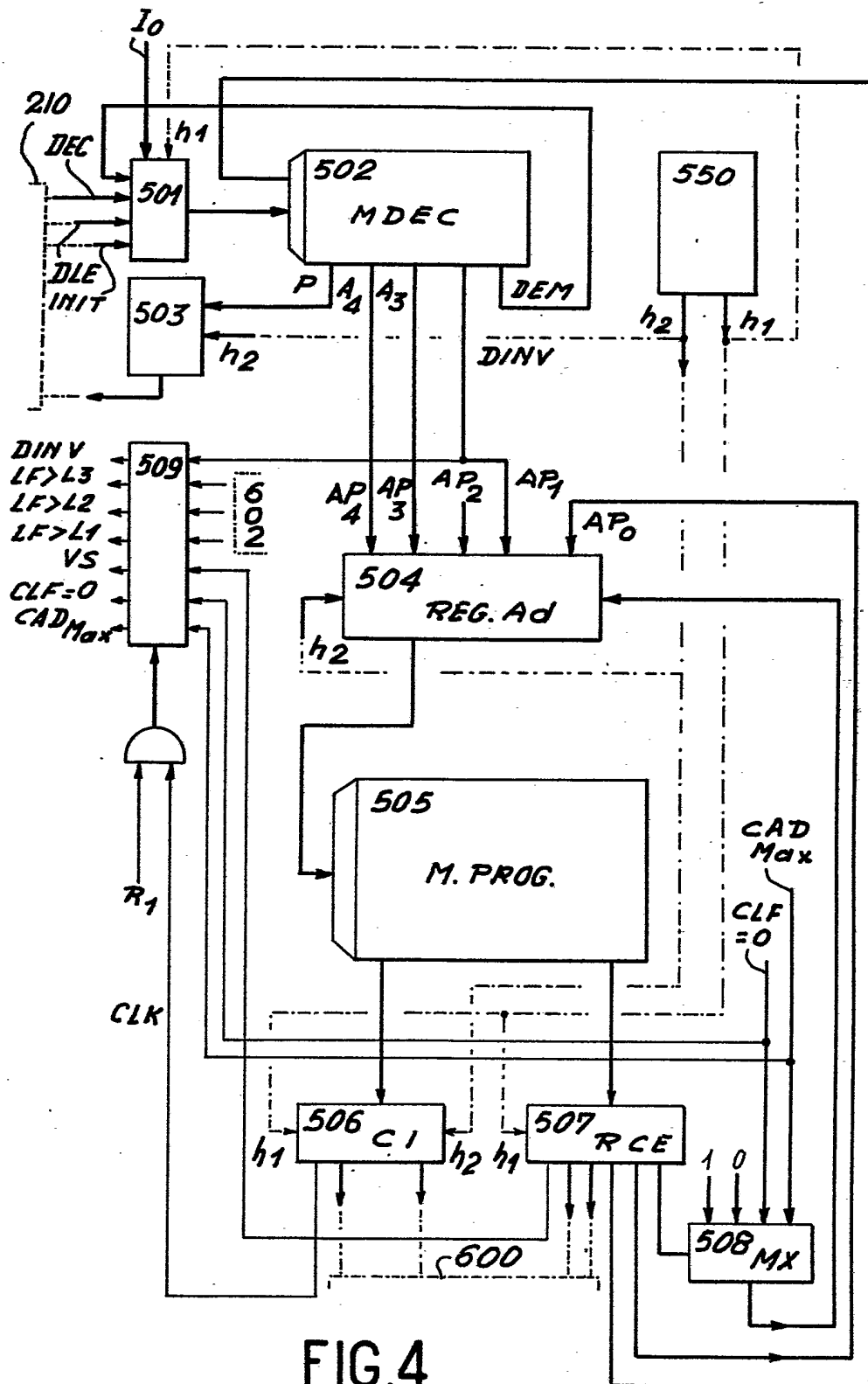
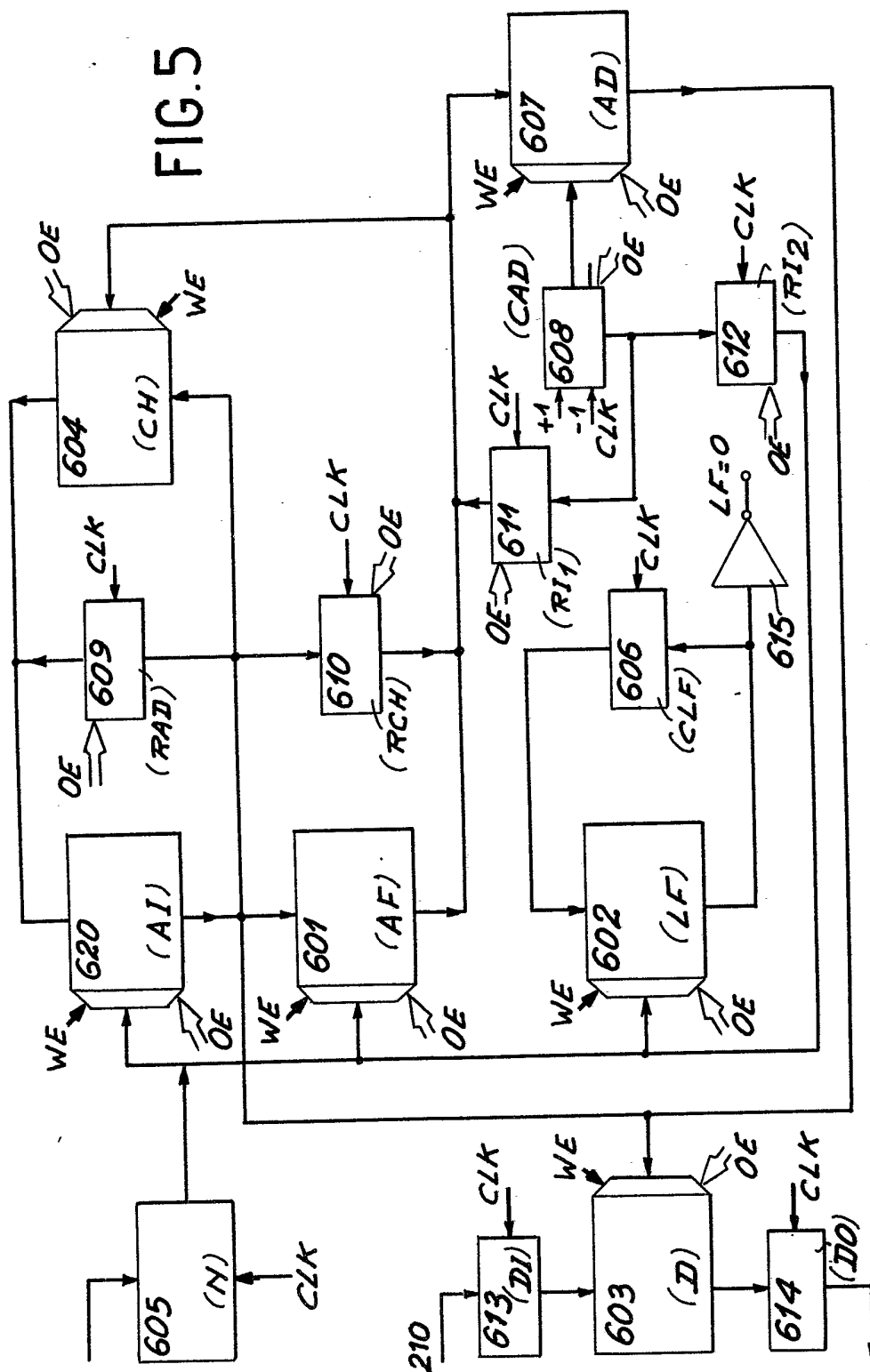


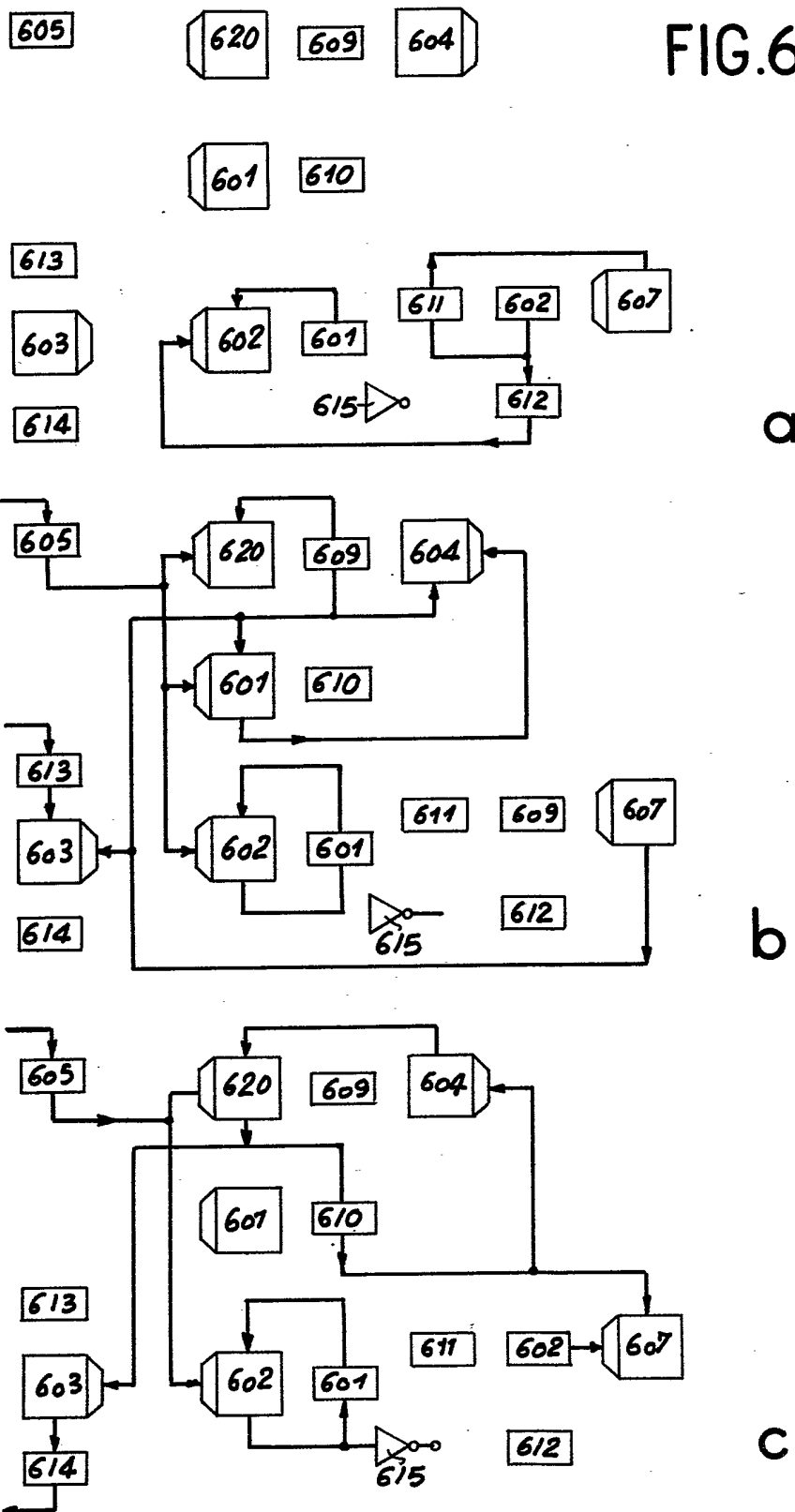
FIG. 4

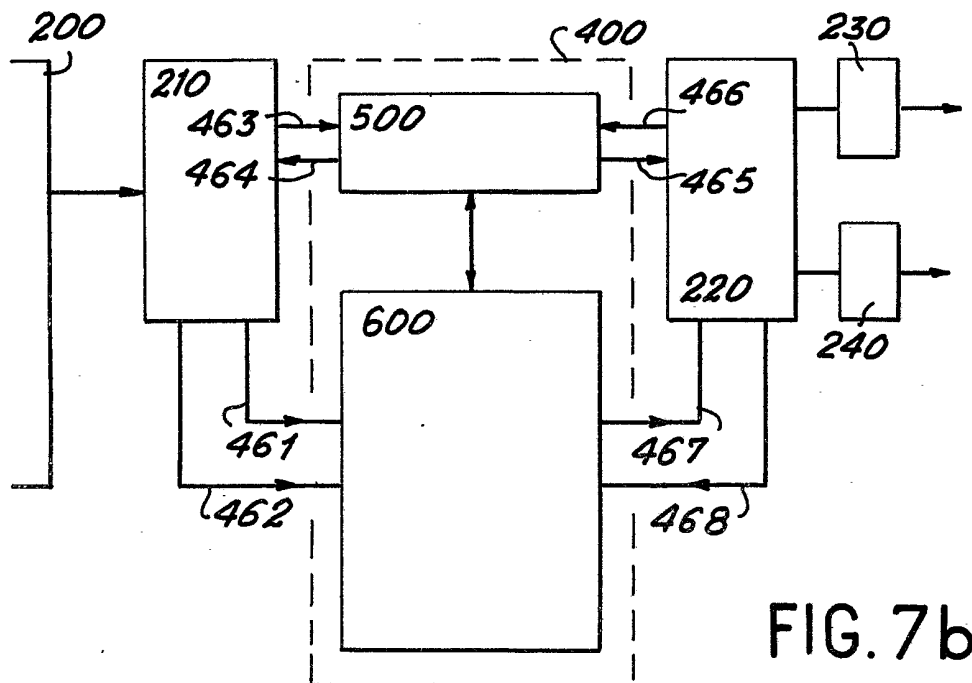
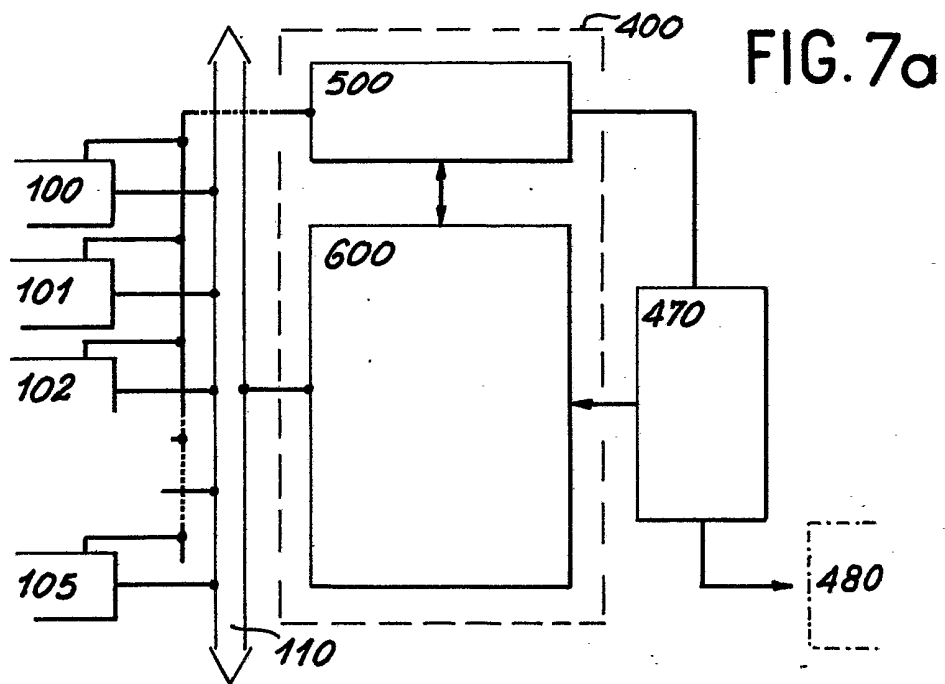
FIG. 5



6 / 10

FIG. 6





8 / 10

FIG. 8

<i>ad</i>	<i>D</i>	<i>CH</i>
0	3	7
1		
2	1	10
3	2	11
4	3	
5		
6	2	9
7	3	4
8	1	2
9	2	
10	1	
11	2	6

