



⑫ FASCICULE DU BREVET A5

616 252

⑳ Numéro de la demande: 15752/75

㉒ Date de dépôt: 03.12.1975

㉓ Priorité(s): 04.12.1974 FR 74 39592

㉔ Brevet délivré le: 14.03.1980

㉕ Fascicule du brevet publié le: 14.03.1980

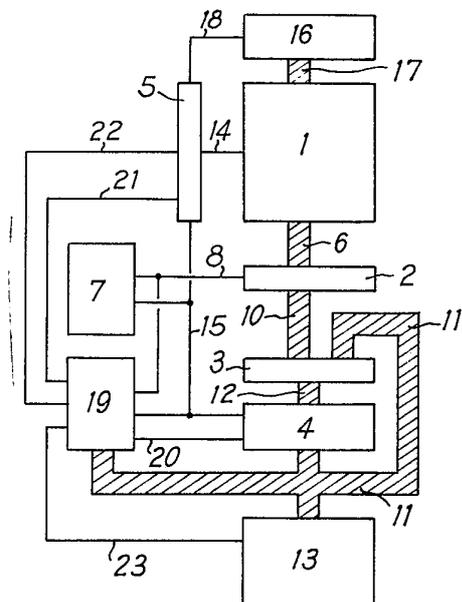
㉗ Titulaire(s):  
ANVAR - Agence Nationale de Valorisation de la Recherche, Neuilly-sur-Seine (FR)

㉘ Inventeur(s):  
Jean F. Suchard, Paris (FR)  
Jacques Vidalin, Bordeaux (FR)  
Hong Hoang Quang, Paris (FR)

㉙ Mandataire:  
E.O. Punschke, Meilen

⑤④ Dispositif de traitement d'un flot d'information et procédé pour la mise en action.

⑤⑦ Le flot d'information reçu d'une source d'information (7) est sous forme séquentielle par élément d'information. On veut éliminer la nécessité d'emmagasiner temporairement une partie importante des données dans une mémoire en vue de l'exécution des opérations de comparaison. A cet effet on effectue de façon séquentielle et directement la comparaison globale (6, 11) de chacun des éléments d'information issus de la base de données en parallèle avec les éléments d'information correspondants des données de référence, ces éléments d'information étant de taille quelconque et pouvant être réduits à l'unité d'information ou bit. Les entités de référence peuvent être de taille fixe, ou de taille variable. Le dispositif comprend une mémoire à lecture-écriture (1) contenant les données de référence, deux opérateurs (2,3) effectuant des fonctions de comparaison, un registre de travail (4) contenant les résultats et un registre-compteur (5). Les liaisons (6 et 11) ont une capacité de 48 à 64 bits. Le dispositif s'utilise en documentation automatique.



## REVENDEICATIONS

1. Dispositif de traitement d'un flot d'information reçu d'une source d'informations sous forme séquentielle par élément d'information comportant:

– une première mémoire (1) destinée au stockage d'entités logiques de référence, avec lesquelles les éléments d'information seront comparés ladite mémoire comportant des cases comportant plusieurs éléments de stockage ordonnés destinées respectivement à contenir un élément d'information d'entités logiques stockées différentes, et la première mémoire ayant une sortie de lecture (6);

– un opérateur d'écriture (16) destiné à un ensemble d'entités logiques de référence dans ladite première mémoire avec les éléments d'information desdites entités logiques de référence sous forme séquentielle dans des cases respectives de ladite première mémoire, en commençant avec la première de ces cases;

– un opérateur de lecture (5) pour commander la sortie de lecture (6) de ladite première mémoire (1) afin de provoquer la lecture séquentielle des contenus de ladite première mémoire case par case de façon non cyclique

– un premier opérateur logique (2) pour réaliser la comparaison simultanée et indépendante de chaque élément d'information reçu avec les éléments d'information lus dans une seule case de stockage par ladite sortie de lecture de ladite première mémoire;

– un second opérateur (3, 4, 25) ayant un registre de sortie temporaire et ayant une entrée reliée à la sortie dudit premier opérateur logique (2) pour combiner la sortie dudit premier opérateur logique avec les contenus dudit registre de sortie et, ensuite, substituer le résultat de la combinaison au contenu du registre de sortie sans autre combinaison avant la réalisation d'une autre comparaison dans ledit premier opérateur logique (2), et

– un organe logique de commande (19) que délivre les signaux nécessaires à l'initialisation et à la commande desdits opérateurs logiques, de l'opérateur de lecture et du registre en réponse à des signaux d'horloge dérivés de ladite source d'information (7) pendant l'écoulement des éléments d'information issus de cette source.

2. Procédé de mise en action du dispositif selon la revendication 1, caractérisé par le fait que l'on sélectionne dans ledit opérateur d'écriture (16) selon les objectifs de recherche, en tant que mots d'information de référence, plusieurs entités logiques de référence

– on enregistre lesdites plusieurs entités logiques de référence dans un tableau à deux dimensions dans ladite première mémoire, dont les colonnes sont destinées à recevoir les entités logiques, les lignes étant destinées à contenir des éléments d'information de chaque entité logique de référence;

– on reçoit séquentiellement sous forme électrique lesdits éléments d'information constituant ledit flot d'information transmis par la source d'information;

– on détecte le premier élément d'information de la première entité logique reçue;

– on compare ledit premier élément d'information simultanément et indépendamment avec les contenus de la première rangée dudit tableau;

– on combine en parallèle les résultats de la première étape de comparaison avec un ensemble prédéterminé d'état initial d'un registre de résultats selon un second ensemble de fonctions logiques choisi en fonction des objectifs de recherche et on enregistre de façon temporaire les résultats de cette étape de combinaison dans ledit registre de résultats;

– on compare l'élément d'information suivant reçu, simultanément et indépendamment, avec le contenu de la seconde rangée dudit tableau;

– on combine en parallèle les résultats de cette seconde étape de comparaison avec les résultats enregistrés de façon temporaire de la première étape de comparaison et on enregistre de façon temporaire les résultats de cette seconde étape de combinaison à la place des résultats enregistrés d'abord dans ledit registre de résultats;

– on poursuit cette suite alternée d'étapes de comparaison et de combinaison jusqu'à, et y compris, soit les étapes de comparaison et de combinaison suivant la détection de dernier élément d'information dudit premier élément logique reçu, soit l'étape de comparaison utilisant la dernière rangée dudit tableau et l'étape de combinaison suivante, selon ce qui se produit d'abord;

– on fournit les résultats de la dernière étape de combinaison comme résultats du traitement suivant de recherche, et

– on procède selon le même processus après la première étape d'enregistrement pour traiter les entités logiques reçues ensuite en commençant à chaque fois par la détection de leur premier élément d'information.

3. Dispositif de traitement selon la revendication 1, caractérisé en ce que le registre de sortie (4) dudit second opérateur logique (3) est une unité distincte ayant une entrée reliée à la sortie dudit second opérateur logique, une sortie reliée à l'entrée dudit second opérateur logique et une autre sortie constituant une sortie dudit dispositif de traitement.

4. Dispositif de traitement selon la revendication 1, caractérisé en ce qu'il comporte des mémoires additionnelles de même type que ladite première mémoire de manière à fournir, avec ladite première mémoire, plusieurs mémoires (1a, 1n) pour enregistrer respectivement différents ensembles d'entités logiques de référence (figures 12 et 13), et en ce que l'organe logique de commande (19) comporte des moyens pour sélectionner une desdites mémoires (1a, 1n) selon un plan prédéterminé.

5. Dispositif de traitement selon la revendication 1, caractérisé en ce que le premier et le second opérateur logique (2, 3, 4) forment ensemble un dispositif logique unique (30) ayant un registre de sortie (4) et ayant trois entrées d'information reliées respectivement à ladite source d'information, à ladite sortie de lecture de la première mémoire et audit registre de sortie (4).

6. Dispositif selon la revendication 5, caractérisé en ce qu'il comporte un opérateur logique additionnel (101) ayant un registre de sortie (100) pour combiner les résultats successifs à la sortie du registre de sortie (4) et des opérateurs logiques supplémentaires (102, 103, 104, 105) pour tester la sortie dudit registre de sortie (100) et engendrer des signaux de commande de décision.

7. Dispositif selon la revendication 5, caractérisé en ce qu'il comporte en outre des moyens de mémorisation tampon (figure 20) pour enregistrer temporairement des parties du flot d'information reçu, ces moyens comportant une mémoire tampon (350), un registre d'adresse (351), et un ou plusieurs registres-doublures (352), lesdits registres d'adresse et doublures étant connectés en boucles (353, 354) à l'entrée d'adressage de ladite mémoire tampon (350) et ayant respectivement des entrées de commande d'écriture (356, 355) pour copier chacun des registres dans l'autre, et ledit registre d'adresse (351) ayant en outre une entrée de commande d'initialisation (357) et une entrée de commande de progression (358).

8. Dispositif selon la revendication 7, caractérisé en ce qu'une partie des registres-doublures sont remplacés par des registres à décalage du type file.

9. Dispositif selon la revendication 7, caractérisé en ce qu'une partie des registres-doublures sont remplacés par des registres du type pile.

10. Dispositif selon revendication 7, caractérisé en ce qu'il comporte en outre des registres-doublures respectivement pour le registre de sortie (4) et l'opérateur de lecture (5) pour

enregistrer de façon temporaire l'état dudit registre de sortie et l'état dudit opérateur de lecture en cas de reprise.

11. Dispositif selon la revendication 7, caractérisé en ce que lesdits moyens de mémorisation temporelle comportent des mémoires additionnelles pour enregistrer les éléments définissant des modifications à faire à un enregistrement contenu dans ces moyens de mémorisation tampon (350) et comportent en outre des moyens de décision (355) agissant en réponse à la sortie dudit registre de sortie (4) pour effectuer, à l'emplacement desdits moyens de mémorisation tampon spécifiés par le contenu d'un des registres-doublures, les modifications définies par lesdits éléments.

12. Procédé selon la revendication 2, selon lequel le traitement à effectuer sur ledit flot d'information transmis et reçu comprend plusieurs sous-processus, caractérisé en ce que l'on sélectionne les entités logiques sous forme de mots d'information de référence en fournissant, comme mots d'information de référence, plusieurs ensembles d'entités logiques de référence chacun ayant un format homogène et tous étant des suites d'éléments d'information;

on enregistre lesdits ensembles homogènes d'entités logiques de référence dans un nombre de tableaux de dimensions appropriées selon leurs formats respectifs et leur nombre,

on détecte le premier et le dernier élément de chaque entité logique reçue indépendamment et en même temps en fonction desdits différents formats respectifs,

les autres étapes dudit processus étant simultanément et individuellement réalisées pour chaque sous-processus, en utilisant pour chacun desdits sous-processus le tableau approprié, le premier et le second ensembles de fonctions logiques, et

on fournit les résultats en fournissant indépendamment les résultats respectifs des sous-processus.

13. Procédé selon la revendication 1, selon lequel le flot d'information reçu de la source contient des séquences distinctes d'entités logiques consécutives, la première et la dernière entités logiques d'une séquence étant reconnaissables en tant que telles, caractérisé en ce que:

on fournit un troisième ensemble de fonctions logiques, choisi selon les objectifs de recherche, pour réaliser une combinaison en parallèle de résultats de la première étape de comparaison avec un ensemble prédéterminé d'état initial d'un registre de résultats,

on détecte le début et la fin de chaque séquence d'entité logique reçue,

à la détection de la fin de chaque entité logique reçue de ladite séquence, on combine en parallèle les résultats de la dernière étape de combinaison dans le processus pour l'entité logique a juste été reçue avec la sortie enregistrée des derniers résultats de combinaison pour une ou plusieurs quelconques entités logiques préalablement reçues de la même séquence, et

à la détection de la fin de la séquence, on fournit des résultats de ladite combinaison au second niveau des résultats de la dernière entité logique de la séquence avec les résultats des entités logiques préalables, comme résultats du traitement de la séquence reçue.

La présente invention concerne un dispositif de traitement d'un flot d'information reçu d'une source d'informations sous forme séquentielle par élément d'information, et un procédé pour la mise en action.

On entend par source d'informations par exemple: des fichiers ou des ensembles de fichiers quel que soit leur support technologique; ou des moyens quelconques de transmission de données; ou des systèmes d'acquisition de données par mesure de paramètres; ou d'une unité de traitement; etc . . .

Le terme «données de référence», qui peuvent être du type logique, numérique ou alphanumérique, est représenté sous une forme compatible avec celle des données issues de la base de données.

On entend par «rapprochement» ou «traitement» l'opération de reconnaissance d'une relation entre telle donnée issue de la base de données et les données de référence. Cette relation peut être absolument quelconque, telle que: identité, implication, plus grand, plus grand ou égal, plus petit, plus petit ou égal, évaluation d'une distance selon une mesure convenable, etc . . .

Ce problème général de rapprochement se rencontre par exemple dans le domaine de la documentation dite automatique, celui de l'analyse syntaxique des langages algorithmiques et des langages de commande, celui du traitement des en-têtes et des codes de procédure en transmission de données, celui de la reconnaissance des formes, celui de contrôle de processus notamment industriels, etc . . .

Selon un premier genre de procédé connu, ce rapprochement est effectué par une suite de comparaisons et/ou d'opérations élémentaires organisées selon un processus arborescent.

Le principe général de ce procédé connu sera rappelé ici pour une meilleure compréhension à l'aide de l'exemple particulier suivant:

soit à comparer un mot délivré par une base de données constituée d'un fichier aux différents mots emmagasinés dans les données de référence, et définissant la question posée audit fichier.

Par exemple, soit à comparer le mot:

«MEDICAMENT»

à l'ensemble de mots:

«RESPONSABILITE CIVILE (DES) MEDECINS (DES) HOPITAUX PUBLICS», en faisant abstraction des mots entre parenthèses.

Les comparaisons élémentaires suivantes seront alors successivement effectuées:

COMPARAISON	REPONSE
M à R	non
M à C	non
M à M	oui
E à E	oui
D à D	oui
I à E	non
M à H	non
M à P	non

et ainsi de suite pour les mots suivants délivrés par le fichier.

On voit donc que le nombre de comparaisons élémentaires qui doivent être effectuées pour l'exploitation du fichier est au moins égal au produit du nombre des mots contenus dans ce fichier par le nombre de mots que comporte la question.

Or une question peut facilement en demander huit. Ainsi, en documentation juridique, elle peut aller jusqu'à vingt (expérience de la Cour de Cassation italienne). On est par conséquent conduit, dès qu'un fichier devient important, à le découper en éléments plus rapidement exploitables, ce qui n'est pas parfois sans soulever des problèmes importants d'adressage et de renvoi.

Mais surtout, dans un tel système de comparaison, le temps lui-même de traitement reste aléatoire. Il résulte de la configuration du mot qui se présente du fichier, en regard de la configuration et de la place même des mots qui sont donnés en question; en effet:

— Le temps de traitement est fonction de la configuration des mots:

Par exemple, pour le mot «MEDICAMENT», comparé à «RESPONSABILITE CIVILE MEDECINS HOPITAUX

PUBLICS», on a, on l'a vu, huit comparaisons élémentaires.

Pour le mot «VETERINAIRE», toujours comparé avec le même ensemble de mots, on aurait:

COMPARAISON	REPONSE
V à R	non
V à C	non
V à M	non
V à H	non
V à P	non

soit cinq comparaisons élémentaires.

— Le temps de traitement est encore fonction de la place que prennent les mots dans le question, puisqu'on passe, dès qu'un mot est reconnu, au mot suivant du fichier.

Pour le mot «CIVILE» comparé au même ensemble de mots, on aurait:

COMPARAISON	REPONSE
C à R	non
C à C	oui
I à I	oui
V à V	oui
I à I	oui
L à L	oui
E à E	oui

soit sept comparaisons élémentaires.

Pour le mot «PUBLICS» toujours comparé avec le même ensemble de mots, on aurait:

COMPARAISON	REPONSE
P à R	non
P à C	non
P à M	non
P à H	non
P à P	oui
U à U	oui
B à B	oui
L à L	oui
I à I	oui
C à C	oui
S à S	oui

Soit onze comparaisons élémentaires.

Or c'est précisément parce que le temps de traitement, 15 non seulement pour chaque mot, mais aussi pour chaque caractère ou octet, reste aléatoire, qu'en documentation automatique on ne peut régler le traitement d'un fichier sur la cadence où les données sont délivrées par les supports, et notamment par les disques ou les mémoires de masse. Les conséquences les plus évidentes en sont la lourdeur des procédures d'entrée-sortie et un encombrement excessif des informations dans la mémoire centrale en regard de la vitesse de traitement.

Outre la procédure de l'exemple ci-dessus, il existe d'autres procédures du même caractéristiques.

Pour pallier les inconvénients de ces procédés connus, on a proposé d'utiliser des dispositifs dénommés «mémoires associatives» ou «à recherche par le contenu». Le principe général d'utilisation de ces dispositifs consiste à effectuer, simultanément et en parallèle, la comparaison d'un élément d'information avec les contenus de toutes les cases d'une telle mémoire, le résultat étant soit le numéro, soit le contenu des cases pour lesquelles la comparaison donne une réponse positive. Ceci implique nécessairement que la logique câblée réalisant l'ensemble de toutes ces nombreuses comparaisons, intégrée dans la structure même de la mémoire, est lourde, complexe, chère et totalement rigide puisqu'on ne peut modifier ni le format des données traitées, ni la nature des comparaisons effectuées.

En pratique, ces mémoires sont finalement peu utilisées en dehors d'applications nobles bien spécifiques. En effet, si l'on désire effectuer des traitements sur des volumes importants de données, on parvient à la réalisation de matériels volumineux et donc chers. Pour réduire ce volume de matériel, on a parfois utilisé des solutions de compromis en combinant ces mémoires avec des processus de recherche arborescents du type ci-dessus, mais en retombant alors des inconvénients des ces derniers.

La présente invention a pour but de résoudre le problème général exposé ci-dessus mais en éliminant tous les inconvénients des procédés antérieurement connus mentionnés précédemment, et notamment la nécessité d'emmagasiner temporairement une partie importante des données issues de la base de données dans une mémoire à aléatoire ou à accès associatif en vue de l'exécution des opérations de comparaison.

L'objet défini dans les revendications permettent d'attendre ce but.

On supposera dans la suite que les données de référence peuvent être considérées comme formées d'un certain nombre d'entités indépendantes, que l'on conviendra d'appeler «entités logiques» de référence.

Ces entités logiques peuvent être de taille fixe, c'est-à-dire composées d'un nombre déterminé d'éléments ou d'unités d'information, comme c'est le cas des caractères, mots, ou codes formés d'un nombre défini de caractères.

Elles peuvent également être de taille variable. On peut alors en général les considérer comme formées d'un nombre quelconque de caractères, eux-mêmes de taille fixe, ce nombre restant inférieur à une certaine limite. Elles sont alors séparées les unes des autres par un caractère particulier, appelé blanc ou espace pour les données alphanumériques qui ont cette structure, caractère qui sera traité comme le dernier caractère de l'entité logique précédente.

On supposera également qu'au moins un partie des informations issues de la base de données a une structure compatible avec celle des données de référence, c'est-à-dire pourra être considérée comme formée d'«entités logiques» de même type que les entités logiques de référence.

D'autres caractères particuliers, dits aussi «caractères spéciaux», sont utilisés pour délimiter des suites d'entités logiques, ou séquences, et éventuellement les caractériser.

On exposera ci-après, pour plus de clarté, l'application de ce procédé très général à l'exemple particulier déjà pris ci-dessus.

En ce qui concerne la phase a) du procédé, les données de référence déjà citées:

«RESPONSABILITE CIVILE (DES) MEDECINS (DES) HOPITAUX PUBLICS» seront considérées comme formées de cinq entités logiques de taille variable, en ignorant les articles entre parenthèses. Le tableau de référence sera dans le cas où l'élément d'information choisi est le caractère:

	R	C	M	H	P
	E	I	E	O	U
55	S	V	D	P	B
	P	I	E	I	L
	O	L	C	T	I
	N	E	I	A	C
	S	□	N	U	S
60	A	-	S	X	□
	B	-	U	U	-
	I	-	-	-	-
	L	-	-	-	-
	I	-	-	-	-
65	T	-	-	-	-
	E	-	-	-	-
	□	-	-	-	-

tableau dans lequel le signe « $\leftarrow$ » représente le caractère de séparation des entités logiques, et le signe « $\rightarrow$ » un caractère quelconque.

Ce tableau pourra également être tronqué de la façon indiquée par la ligne horizontale entre la douzième et la treizième ligne si on décide que les entités logiques de référence sont limitées à 12 caractères.

Dans le cas où l'on choisit de prendre le bit comme élément d'information, les caractères étant codés sur huit bits, en ASCII avec parité paire, le début du tableau aura alors la forme suivante:

```

0 1 1 0 0
1 1 0 0 0
0 0 1 0 0
0 0 1 1 0
1 0 0 0 1
0 0 0 0 0
1 1 1 1 1
1 1 0 0 0
1 1 1 1 1
0 0 0 1 0
1 0 1 1 1
0 1 0 1 0
0 0 0 0 1
0 0 0 0 0
1 1 1 1 1
1 1 1 1 0
etc . . .

```

Le tableau comporte alors quatre-vingt seize ou cent vingt lignes, selon que les entités logiques seront limitées à 12 caractères ou auront 15 caractères.

A d'autres éléments d'information correspondront des tableaux ayant le même nombre de colonnes et un nombre différent de lignes.

En ce qui concerne la phase c) du procédé, mise en oeuvre avec une première fonction de comparaison simple «identique-différent» appliquée au niveau du caractère, on compare le premier caractère de l'entité logique reçue avec les caractères contenus dans la première ligne du tableau, soit : R C M H P et on note le résultat qui peut être ici uniformément Faux, ou comporter un résultat Vrai, au cas où le caractère reçu est l'un des cinq caractères différents contenus dans cette ligne.

Dans le cas où l'on effectue la même comparaison bit par bit, la première ligne du tableau est 0 1 1 0 0, et le résultat de cette première comparaison sera identique à cette ligne 0 1 1 0 0 ou égal à son complément 1 0 0 1 1, selon que le premier bit de l'entité à rapprocher est un 1 ou un 0. En effet, on effectue alors l'une des opérations de comparaison suivantes:

```

0 1 1 0 0    ou    0 1 1 0 0
1 1 1 1 1          0 0 0 0 0
-----          -----
0 1 1 0 0          1 0 0 1 1

```

le 1 correspondant à VRAI et le 0 à FAUX.

En ce qui concerne la phase d) du procédé, on effectue la seconde comparaison avec la deuxième ligne du tableau E I E O U et cette comparaison peut donner ici aucun, un, ou deux résultats VRAI, puisque cette ligne contient deux caractères identiques.

La seconde fonction de combinaison sera ici une opération logique «ET», qui ne peut donner un résultat VRAI que si les deux premiers caractères reçus forment un des bigrammes : RE, CI, ME, HO ou PU.

Dans le cas du traitement bit par bit, le résultat de la deuxième comparaison est 1 1 0 0 0 ou son complément

0 0 1 1 1, et la combinaison par un ET logique donnera, selon des deux premiers bits de l'entité logique à rapprocher:

	Premier bit	Second bit	Résultat partiel pour les deux premiers bits
5	0	0	0 0 0 1 1
	1	0	0 0 1 0 0
	0	1	1 0 0 0 0
10	1	1	0 1 0 0 0

En ce qui concerne la phase e) du procédé, dans le cas du traitement caractère par caractère, la poursuite de l'opération jusqu'au rang n ne donne un résultat non globalement Faux que si la chaîne des n caractères reçus est identique à celle contenue dans une des colonnes du tableau, tronqué au rang n. Si on poursuit le traitement jusqu'au dernier caractère «espace» de l'entité reçue, ou jusqu'à la dernière ligne d'un tableau complet, le résultat de rapprochement n'est non globalement Faux que si l'entité logique reçue est identique à une des entités logiques de référence.

Le résultat final est le même pour un traitement bit par bit, au prix d'un nombre d'itérations huit fois plus grand, mais portant sur des opérations plus simples.

Le procédé met en oeuvre deux fonctions différentes qui seront convenablement choisies selon le résultat recherché, et qui dépendent également de la taille des éléments d'information manipulés.

Dans le cas le plus simple où le traitement est effectué bit par bit, la première fonction sera le plus souvent, comme dans l'exemple ci-dessus, une simple fonction booléenne IDENTITE, appliquée entre l'élément incident et les éléments contenus dans la ligne de même rang du tableau, donnant ainsi un résultat qui peut s'écrire :

$R = x = X$   
x étant l'élément incident, et X représentant le vecteur booléen ligne du tableau.

Cette fonction est utilisable pour les rapprochements de type égal-différent, portant sur des entités de taille fixe ou variable.

D'autres fonctions logiques ou arithmétiques peuvent être employées en vue de réaliser des rapprochements plus complexes, du type «plus grand que», «plus grand ou égal», ou «écart arithmétique».

La seconde fonction complète la définition du rapprochement obtenu. Il s'agit toujours d'une opération parallèle, effectuée entre deux ensembles d'éléments d'information comportant le même nombre d'éléments, élément par élément et sans propagation latérale de reports. Comme pour la première fonction, on peut effectuer de nombreuses opérations logiques ou arithmétiques, mais la fonction la plus employée sera la fonction logique ET, comme dans l'exemple ci-dessus, qui permet de vérifier que la condition exprimée par la première fonction est bien remplie par tous les éléments d'information de l'entité logique examinée. En particulier, l'association de cette seconde fonction avec la première fonction IDENTITE donne un rapprochement «égal-différent» portant sur un nombre quelconque d'entités logiques de référence de taille quelconque.

On peut écrire alors en général :  $S := S \wedge R$ , où R est le résultat de la première fonction. La dernière valeur de S donne le résultat du rapprochement.

Ceci suppose que le vecteur S a été initialisé à VRAI partout et que les entités logiques de taille variable sont terminées par le caractère de séparation, y compris à l'intérieur du tableau.

Un rapprochement selon la distance de Hamming pourra être effectué à l'aide de la première fonction «OU EXCLUSIF» et d'une seconde fonction de type comptage. Il s'agit en effet de

compter pour chaque entité logique de référence le nombre de bits différents. Le comptage n'a pas besoin d'être poussé jusqu'à un ordre élevé, mais il ne doit pas être effectué en mode modulo, pour éviter toute ambiguïté. Un ensemble de compteurs non cycliques à quatre états donne en général un résultat satisfaisant, en indiquant par un état (0) la réception correcte d'une entité logique de référence, par un état (1) ou (2) cette même réception avec une erreur simple ou double, et par un état (3) une entité logique non reconnue ou reçue avec un nombre d'erreurs supérieurs.

Pour obtenir un même rapprochement, on peut mettre en oeuvre selon le procédé plusieurs associations d'une première et d'une seconde fonction, ou encore une opération unique équivalente à ces deux fonctions sans mise en évidence d'un résultat intermédiaire.

Le procédé s'applique également au traitement d'une séquence d'entités logiques de taille quelconque, cette séquence pouvant être de longueur fixe ou variable. Dans ce cas, le procédé est appliqué successivement et indépendamment à chacune des entités logiques de la séquence.

Selon une autre caractéristique, les résultats des rapprochements successifs effectués au niveau des entités logiques d'une séquence peuvent être stockés en vue d'une exploitation ultérieure.

Ce stockage peut s'appliquer à la totalité desdits résultats, ou être limité à ceux qui sont globalement non nuls et correspondent ainsi à la reconnaissance d'au moins une des entités de référence.

Selon une autre caractéristique, les résultats des rapprochements successifs effectués au niveau des entités logiques d'une séquence peuvent être combinés entre eux selon une troisième fonction choisie de telle sorte que le résultat de cette dernière combinaison représente le résultat du rapprochement pour ladite séquence.

La troisième fonction peut être du même type que la seconde. Une fonction OU permet par exemple de regrouper tous les résultats positifs des rapprochements effectués pour une séquence et définir ainsi les entités reconnues indépendamment de leur ordre.

La troisième fonction peut aussi être d'un type différent, comme un comptage des vecteurs résultats non nuls, qui permet de déterminer le nombre d'entités de référence reconnues avec répétition dans une séquence, ou encore comme la vérification d'un ordre particulier d'occurrence des entités logiques de référence dans une séquence.

Selon une autre caractéristique, les résultats des rapprochements des entités logiques d'une séquence peuvent être successivement rapprochés d'un ensemble de résultats-types de rapprochement, à l'aide d'une deuxième application, simultanée ou non, du procédé de rapprochement tel que décrit plus haut.

On peut effectuer ainsi des tests complexes, tels que d'encadrement, de recherche de l'ordre d'occurrence des entités logiques d'une séquence, ou de reconnaissance d'une situation comme par exemple en contrôle industriel ou en diagnostic médical.

Selon une autre caractéristique, les rapprochements appliqués successivement aux entités logiques d'une séquence peuvent l'être en mettant en oeuvre des tableaux et/ou des fonctions différentes, selon le rang de l'entité logique.

On peut ainsi appliquer directement le procédé à une séquence comportant une succession d'entités logiques de types ou de significations différentes. C'est le cas en particulier de séquences comportant un ou plusieurs caractères d'identification suivis de données logiques ou numériques, comme on en rencontre dans les procédures de transmission de données, les langages de programmation et de commande, et le contrôle industriel.

Bien entendu, on peut appliquer le procédé à une suite de séquences, conformément aux différentes possibilités mentionnées juste ci-dessus en ce qui concerne le traitement d'une séquence.

Selon une autre possibilité, l'application du procédé de rapprochement à une séquence particulière est conditionnée et définie par des résultats de rapprochements effectués préalablement.

Selon cette disposition, notamment, n'importe quelle entité logique ou séquence d'entités logiques peut être utilisée comme délimiteur de séquence, ou d'ensemble de séquences tel qu'article, sous-article, fichier, sous fichier, ou toute subdivision des informations contenues dans la base de données et peut être utilisée éventuellement pour spécifier la nature et les paramètres du ou des rapprochements ou des transferts à appliquer aux entités logiques desdites subdivisions. On obtient ainsi une très grande souplesse dans le traitement de structures complexes de données, limitée seulement par la complexité des décisions logiques à effectuer. Cette propriété présente également l'intérêt de faciliter considérablement l'application du procédé à des fichiers ou bases de données existantes, en évitant leur restructuration.

Dans le cas où l'on a le choix des structures de données, il est souvent préférable, et toujours plus simple, d'utiliser comme délimiteurs et spécificateurs des caractères particuliers ou «caractères spéciaux» réservés exclusivement à cet usage. En effet, selon une autre caractéristique de l'invention, l'identification de ces caractères est réalisée par une application simultanée et indépendante du procédé de rapprochement, à tous les caractères constituant les données à traiter, considérés comme des entités logiques de taille fixe, le résultat de cette application étant utilisé pour contrôler les différentes phases du procédé.

En particulier, ces caractères spéciaux peuvent contrôler :

- 1) L'initialisation de l'application principale du procédé, par chargement de la ou des tables avec les entités logiques de référence à partir d'une source convenable qui peut être la base de données.

- 2) Le début d'une opération de rapprochement.

- 3) Le choix des paramètres dudit rapprochement (table et fonction).

- 4) La fin d'une opération de rapprochement.

- 5) La définition des données transférées vers un autre organe du système en cours ou à l'issue de l'opération de rapprochement; cette dernière fonction de commande est la plus complexe car ces données à transférer peuvent être de nature et d'origine très diverses. En effet, il peut s'agir : des résultats du rapprochement de chaque entité logique d'une séquence; d'une combinaison plus ou moins complexe de résultats de rapprochements; d'une partie du contenu de la ou des tables, ou d'une table annexe associée à la table contenant les données de référence; d'un ou plusieurs séquences d'entités logiques issues de la base de données. Dans ce dernier cas, la définition des données transférées peut mettre en jeu plusieurs caractères spéciaux, qui définissent chacun un élément de la spécification : séquence soumise au rapprochement avec transfert conditionnel, condition à remplir pour que ledit transfert soit effectivement exécuté, séquence soumise à un rapprochement avec stockage des résultats, séquence ou ensemble de séquences non soumises à rapprochement mais dont le transfert est conditionné par un ou plusieurs résultats de rapprochement, paramètres des divers rapprochements et de décision. Il est ainsi possible de retrouver en particulier la plupart des résultats obtenus ou proposés avec les différents types de mémoires associatives connus jusqu'alors, avec les limitations d'une organisation partiellement série, mais avec une économie et une souplesse d'application très supérieure et une vitesse en général suffisante, parfaitement adaptée au débit des supports d'information capables de fournir de façon continue des flots importants de données.

6) Le blocage temporaire de la recherche par rapprochement des caractères spéciaux; cette disposition est importante dans tous les cas où l'usage exclusif des caractères spéciaux comme délimiteurs est impossible à respecter, notamment lorsque la base de données contient des zones successives de structures différentes, par exemple une structure de mot pour des données logiques ou numériques à l'intérieur d'un fichier alphanumérique. Le blocage peut s'appliquer: sur une étendue déterminée jusqu'à la fin de l'enregistrement physique; ou plus généralement en étant limité par tout délimiteur indépendant des données elles-mêmes. En l'absence de dispositif convenable, on peut utiliser un des procédés connus de transparence des procédures de transmission de données, mais cela conduit en général à des solutions plus lourdes ou moins sûres.

Le procédé, tel qu'il est décrit ci-dessus, est applicable au traitement de tous les fichiers qu'on rencontre en documentation automatique.

Soit, en effet, un tel fichier: il se compose nécessairement d'un certain nombre d'articles ou suites de séquences; de sous-articles ou séquences; de mots clés ou entités logiques. Ces suites de séquences, ces séquences et ces entités logiques peuvent être de longueur fixe, ou, le plus généralement, de longueur variable et sont alors définies par leurs en-têtes ou délimiteurs ou, pour les entités logiques, par le caractère blanc ou espace qui les précède, à défaut d'un autre caractère spécial.

Dans cette application particulière, ces en-têtes ou délimiteurs sont des entités logiques spéciales. Des entités logiques spéciales de référence leur correspondent sur une table; et le résultat des rapprochements successifs effectués sur cette table commande l'exécution des diverses fonctions du procédé de rapprochement. Un sélecteur permet de déterminer la fonction à exécuter selon l'entité logique spéciale qui a été reconnue. Les codes, permettant de sélectionner ces différentes entités spéciales en fonction de l'opération qu'on veut leur voir commander, sont fournis par programme à partir de l'unité principale de traitement.

Ces entités logiques spéciales commandent l'une des opérations suivantes:

- 1) Une opération de rapprochement pour les entités logiques de la séquence définie par l'entité spéciale reconnue.
- 2) Une opération de stockage pour les entités logiques de la séquence définie par l'entité spéciale reconnue. Un caractère spécial placé à la fin de chaque séquence peut évidemment commander la fin d'opération de stockage. Il est le dernier caractère stocké, mais il n'a aucune signification typographique. Il commande, de plus, l'initialisation du rapprochement pour l'entité logique qui suit.

Le traitement des résultats obtenus commande, lorsqu'il est positif, le transfert vers l'unité principale de traitement des entités qui ont été stockées pour même suite de séquence.

Pour des fichiers structurés en longueur fixe, l'enchaînement des diverses opérations peut être commandé de façon purement itérative.

Le code caractérisant le traitement à effectuer sur les résultats de chaque suite de séquences peut comporter l'indication de ce que l'on doit ou non cesser l'exploitation du fichier lorsqu'a été obtenu le résultat recherché. La fin d'exploitation du fichier est alors commandée par l'intermédiaire de l'unité principale de traitement.

Les principaux traitements à effectuer pour chaque suite de séquences sur les résultats de rapprochements obtenus sont les suivants:

- 1) Une fonction OU entre les différents résultats partiels, pour déterminer qu'au moins un des résultats recherchés a été obtenu.
- 2) Une fonction de comparaison entre les différents résultats obtenus et un ensemble de résultats que l'on cherche à obtenir. Il en est ainsi, en particulier, lorsqu'on veut déterminer si toutes

les entités logiques de référence ont été reconnues dans une suite de séquence.

3) Une fonction permettant de déterminer si l'ordre d'occurrence des entités logiques est conforme à l'ordre voulu.

4) Une fonction de comparaison exécutée selon le procédé lui-même est s'effectuant entre, d'une part, les différents résultats obtenus et, d'autre part, plusieurs ensembles de résultats type caractérisant les résultats que l'on cherche à obtenir. Cette fonction de comparaison simultanée peut ou non s'effectuer sur des résultats obtenus selon un ordre choisi d'occurrence, comme il est dit ci-dessus.

A titre d'exemple, soit un fichier bibliographique dont les articles sont structurés comme ci-dessous:

```

15 [ 4 ] AUT [ ] HUGO [ ] VICTOR [ 2 ] TIT [ ] HERNANI [ 2 ]
INDEX [ ] FRANCE [ ] THEATRE [ ] ROMANTIQUE [ 2 ]
DATE [ ] 1840 [ 2 ] EDIT [ ] GARNIER [ 2 ]
20 DATEDI [ ] 19680515 CLDEC [ ] 03458186 [ 2 ]
NUM [ ] 0345 HUG [ 2 ] [ 3 ] [ n. s. ] [ n. s. ] [ 4 ]

```

Un tel article a des en-têtes dont les significations sont les suivantes:

```

AUT [ ] : AUTEUR
TIT [ ] : TITRE
30 INDEX [ ] : INDEXATION
DATE [ ] : DATE PREMIERE PUBLICATION
EDIT [ ] : EDITEUR
DATEDI [ ] : DATE EDITION
CLDEC [ ] : CLASSIFICATION DECIMALE
35 NUM [ ] : NUMERO BIBLIOTHEQUE
[ n. s. ] : CARACTERE SANS SIGNIFICATION OU
TYPOGRAPHIQUE.

```

Un tel article pourrait évidemment comporter d'autres en-têtes ou d'autres séquences, par exemple: nom de celui qui a rédigé la préface ou les diverses notes que comporte l'édition; nom de l'imprimeur au date du dépôt légal, etc . . .

Le signe [ ] signifie que l'on a un caractère blanc ou espace. Il commande l'initialisation du rapprochement pour l'entité logique qui le suit.

Le signe [ 2 ] signifie que l'on a un caractère spécial sans signification typographique et dont la fonction est de commander la fin de l'opération de stockage en même temps que l'initialisation du rapprochement l'entité logique qui le suit.

Le signe [ 3 ] signifie que l'on a un caractère spécial dont la fonction est de commander le traitement des résultats obtenus pour l'article considéré. Il est suivi d'un certain nombre de caractères qui ne commandent aucune fonction et permettent simplement que le traitement des résultats reçus soit exécuté avant le premier rapprochement que demande l'article suivant. Ils s'achèvent nécessairement par un caractère spécial [ 4 ] qui commande la remise à zéro de tous les résultats partiels ou finaux qui ont été obtenus et initialise le rapprochement à effectuer sur la première entité spéciale de l'article qui suit.

Ces différents caractères spéciaux sont enregistrés sur le fichier aux emplacements qui conviennent, par programmation, au moment de la constitution du fichier.

Mais, il est évident que ces fonctions commandées par des caractères spéciaux peuvent l'être par des entités logiques spéciales. Et il faudra tenir compte du moment exact où ces fonctions seront commandées, c'est-à-dire à la fin du rapprochement de ces entités spéciales.

On pourrait alors avoir comme en-têtes :

ART ⊣: DEBUT D'ARTICLE,

dont la fonction est de remettre à zéro tous les résultats partiels ou finaux qui ont été obtenus pour la suite de séquence qui précède.

FART ⊣: FIN D'ARTICLE

qui commande le traitement des résultats reçus. Ce traitement va s'effectuer pendant le rapprochement de l'entité spéciale ART ⊣ qui le suit.

Il en est de même pour le caractère spécial | 2 |. Il peut être remplacé par une entité spéciale. Et celle-ci ne doit avoir aucune signification typographique. Mais, dans cet exemple, l'initialisation du rapprochement peut toujours être commandé par la caractère spécial ⊣.

Le fichier ayant été constitué, l'utilisateur aura le choix de la question qu'il pose et de la réponse qu'il veut obtenir.

Par exemple, il veut obtenir le TITRE des oeuvres du THEATRE de VICTOR HUGO et les NUMEROS sous lesquels ces ouvrages sont classés dans la bibliothèque.

Avant l'exploitation du fichier, il enregistre sur la table réservée aux entités spéciales AUT ⊣, TIT ⊣, INDEX ⊣, NUM ⊣, mais en faisant que le sélecteur, pour les rapprochements effectués sur les entités spéciales AUT ⊣, INDEX ⊣, puisse, en cas de résultat positif, commander une fonction de rapprochement pour la séquence suivante; et que, de même, pour les rapprochements effectués sur les entités spéciales TIT ⊣, NUM ⊣, il puisse, en cas de résultat positif, commander une fonction de stockage de la séquence suivante.

Tout stockage, en vue d'un traitement particulier, peut être commandé par les entités logiques spéciales. Il en serait ainsi, dans l'exemple choisi, si l'utilisateur avait voulu n'avoir pour réponse que les éditions du théâtre de Victor Hugo postérieures à une certaine date, soit 1950. L'entité spéciale DATEDI ⊣ est alors à enregistrer sur la table réservée aux entités spéciales; et lorsqu'elle aura été reconnue par rapprochement, le sélecteur commandera une opération de stockage particulier qui permettra, au moment du traitement des résultats, d'effectuer entre cette information stockée – soit 19680515 – et l'information de référence – soit 19500000 – une opération de comparaison simple en dehors du procédé de rapprochement décrit ci-dessus.

Toujours avant l'exploitation du fichier, l'utilisateur fixera la nature des opérations de rapprochement qu'il veut faire effectuer par le procédé décrit ci-dessus, ici une opération comparaison-identité. Et il fera enregistrer, sur la table réservée aux entités logiques, les entités logiques de référence correspondant à sa question, soit ici: HUGO ⊣, VICTOR ⊣, THEATRE ⊣. L'information de référence 19500000 ne donne pas lieu dans cet exemple à un rapprochement selon la procédure décrit ci-dessus et n'est donc pas enregistrée sur cette table.

Pour qu'il y ait réponse positive à la question posée, il faut que, pour un même article, toutes les trois entités logiques HUGO ⊣, VICTOR ⊣, THEATRE ⊣, aient été reconnues. Cela suppose, si ces trois entités ont été enregistrées dans les trois premières colonnes de la table des entités logiques de référence, que l'on ait un résultat global identique au vecteur de référence

1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

si la table peut contenir seize entités de référence. Et l'utilisateur choisira pour traitement des résultats obtenus au niveau de chaque article, la fonction qui correspond à cette comparaison identité.

L'invention a pour autre objet des modes particuliers d'application du procédé tout particulièrement destinés au traitement de données de longueur variable, telles que celles rencontrées comme exposé ci-dessus en documentation automatique ou en tout autre domaine où l'on a à manipuler une base de données constituée essentiellement de textes en langage naturel.

Ces modes particuliers d'application du procédé mettent en oeuvre un premier procédé qui se caractérise en ce que la reconnaissance d'un caractère spécial, indiquant le début d'une entité logique ou d'une séquence d'entités logiques à rapprocher, provoque l'initialisation des rapprochements portant sur ces entités logiques, l'utilisation des résultats de rapprochement effectués précédemment, l'exécution de fonctions auxiliaires telles que de stockage, et en ce que l'identification d'un caractère spécial ou d'une entité logique spéciale, indiquant la nature de la séquence suivante d'entités logiques, définit la nature des rapprochements effectués, l'utilisation des résultats obtenus lors du rapprochement de cette séquence et la nature desdites fonctions auxiliaires.

En effet, dans une base de données en langage naturel, ou semi-naturel, c'est-à-dire composés de mots issus d'un langage naturel mais utilisés dans le cadre d'une syntaxe plus stricte, les entités logiques sont des mots séparés par des caractères blancs ou espaces. Selon l'organisation de la base de données, certains de ces espaces sont remplacés par des caractères spéciaux différents servant de repères de début de séquences ou de zones. Pour éviter de multiplier le nombre de ces caractères spéciaux, on utilise des entités logiques dites spéciales définissant la nature des séquence et leur signification, telles que:

25 AUT ⊣  
TIT ⊣  
INDEX ⊣ etc . . .

La recherche systématique et simultanée de ces entités logiques spéciales permet d'effectuer les rapprochements convenables sur les séquences d'entités logiques à rapprocher, en utilisant un minimum de caractères spéciaux, éventuellement réduit au seul caractère blanc ou espace.

Dans ce dernier cas, il doit y avoir utilisation absolument exclusive des entités logiques spéciales. Ceci permet d'adapter le procédé à des bases de données de structures différentes, c'est-à-dire comportant un nombre variable de caractères spéciaux, par exemple construites dans le cadre de procédés classiques de gestion de ces bases de données.

La structure inventive permet un fonctionnement simultané des trois, ou quatre, sous-systèmes de traitement, avec un cycle de fonctionnement aussi simple et rapide que celui d'un système de traitement unique.

Par ailleurs, les trois mémoires de table des trois premiers sous-systèmes peuvent être adressés par un dispositif d'adressage commun.

Tous ces organes sont chacun de nature essentiellement modulaire, ce qui permet d'adapter indépendamment la capacité de chaque sous-système aux nombres de caractères ou d'entités logiques de référence, et ou de résultats-types utiles pour un application déterminée.

En cas d'erreur, pour une reprise portant sur une partie quelconque du flot de données issu de la source, toutes les informations caractéristiques nécessaires à cette reprise sont conservées dans des registres de doublure reliés de façon croisée aux registres contenant ces informations de telle sorte que les contrôles par exemple par partié ou code cyclique, effectués sur chacune de ces parties, provoquent à la fin de chacun d'eux, soit la copie du registre principal dans le registre-doublure en cas de contrôle correct, soit la copie du registre-doublure dans le registre principal en cas d'erreur.

La modification du contenu d'un ou plusieurs enregistrements soumis à rapprochement est possible, si la position du des articles ou sous-articles à modifier est repérée lors de la lecture de chaque enregistrement par un ou plusieurs pointeurs conservés jusqu'à la fin de la ladite lecture et qui seront utilisés pour effectuer la ou les modifications dans une mémoire-tam-

pon contenant la totalité de l'enregistrement, celui-ci étant ensuite ré-écrit en totalité sur le support de base de données.

Pour le stockage a priori de séquences ou suites de séquences d'entités logiques soumises en tout ou partie à rapprochement, et susceptibles d'intéresser en tout ou partie l'organe utilisateur, il est possible de conserver systématiquement un pointeur de début lors de la lecture d'une telle séquence ou suite de séquences jusqu'à ce que la décision, résultat de l'ensemble des rapprochements effectués, commande: soit la remise à jour dudit pointeur à la fin de la ou des séquences stockées en vue de la conservation des entités stockées pour transfert ultérieur, soit le retour audit pointeur du mécanisme d'adressage de la mémoire-tampon pour enregistrement d'une nouvelle séquence ou suite de séquences avec destruction de la précédente non souhaitée.

Ces caractéristiques, ainsi que l'ensemble de l'invention, seront plus aisément exposées au cours de la description suivante d'exemple de réalisation de systèmes selon l'invention.

Cette description sera faite en référence aux dessins annexés, sur lesquels:

- la figure 1 représente schématiquement l'organisation d'un système de rapprochement selon l'invention,
- la figure 2 est un schéma du premier opérateur,
- la figure 3 est le schéma d'un exemple de réalisation du deuxième opérateur,
- la figure 4 est un exemple de réalisation du registre de travail,
- les figures 5 et 6 sont des exemples de réalisation du premier opérateur,
- la figure 7 représente schématiquement une variante de réalisation,
- la figure 8 est un exemple de réalisation de l'opérateur de ladite variante,
- la figure 9 représente schématiquement une deuxième variante de réalisation,
- la figure 10 est un exemple de réalisation de l'opérateur de cette deuxième variante,
- la figure 11 est une variante de réalisation combinant en un opérateur unique les deux opérateurs précités et le registre de travail,
- la figure 12 représente une variante du système comportant plusieurs de mémoire,
- la figure 13 représente une variante selon laquelle les différents ensembles de données de référence sont contenus dans différentes parties d'une mémoire commune,
- la figure 14 représente une variante selon laquelle sont prévus plusieurs types de premier opérateur,
- la figure 15 représente un deuxième mode de réalisation du système selon l'invention,
- la figure 16 représente une réalisation de l'organe d'utilisation et de décision que comporte le système,
- la figure 17 représente un système complexe de rapprochement formé de trois sous-systèmes, et
- la figure 18 représente l'organisation logique d'une mémoire particulièrement adaptée à la réalisation des systèmes selon l'invention.
- la figure 19 représente, sous forme de schéma fonctionnel, un mode de réalisation du système selon l'invention, est
- la figure 20 représente le schéma générique d'un mode de réalisation des mémoires-tampons selon l'invention.

Dans un premier mode de réalisation, représenté schématiquement sur la figure 1, le système de rapprochement comporte essentiellement une mémoire à lecture-écriture 1, contenant les données de référence, deux opérateurs 2 et 3 réalisant respectivement la première et la deuxième fonctions, telles que décrites plus haut, un registre de travail 4, contenant les résultats, et un registre-compteur 5.

Les caractéristiques de tous ces organes dépendent principalement de la structure des données de référence, et du mode d'exécution choisi pour l'opération de rapprochement.

Nous allons préciser cela à l'aide de l'exemple déjà utilisé plus haut pour l'exposé du procédé.

Dans cet exemple, les données de référence comportent cinq entités logiques de taille variable, du type chaîne de caractère alphanumériques. La taille maximale de ces entités est de 15 caractères, espace inclus. Nous supposons donc, en majorant un peu, que les données de référence pourront comporter au maximum 8 entités logiques de 16 caractères.

Dans ledit exemple, les opérations de rapprochement selon l'invention ont été décrites comme pouvant être exécutées à deux niveaux différents, au niveau du caractère ou au niveau du bit.

Dans le premier cas, chaque case de la mémoire 1 contiendra une ligne du tableau de référence, c'est-à-dire les caractères de même rang des différentes entités logiques de référence. Ladite mémoire comportera donc 16 cases, dont la capacité sera de 48 à 64 bits selon le code utilisé pour représenter les caractères, soit en tout de 768 à 1024 bits.

La liaison 6 reliant la mémoire 1 au premier opérateur 2 devra avoir un largeur, c'est-à-dire un nombre de signaux transmis, égale à la capacité d'une case, soit 48 à 64 bits. L'opérateur 2 reçoit également des informations de la source 7 par la liaison 8, en série par caractère dans ce cas. L'opérateur 2 comporte donc des circuits capables d'effectuer en parallèle l'opération réalisant la première fonction, telle que définie plus haut, sur 8 caractères.

La figure 2 représente schématiquement la structure correspondante, composée de 8 opérateurs élémentaires 9.1 à 9.8, recevant simultanément par la liaison 8 le caractère à comparer, et séparément par les liaisons 6.1 à 6.8 les caractères de référence issus de la mémoire 1.

Le résultat de cette première opération est transmis au deuxième opérateur 3 par la liaison 10 formée par le regroupement des liaisons de sortie des opérateurs 9.1 à 9.8. Sa composition dépend de la complexité de ce résultat et, par conséquent, de la nature de la première fonction. Pour une opération simple, de comparaison égal/différent, implication ou analogue, l'opérateur élémentaire 9 ne fournira qu'un seul signal de sortie, et la liaison 10 n'aura à transmettre que 8 signaux au second opérateur 3. Pour des opérations de comparaison multiple, telles que celle qu'on peut obtenir avec le circuit intégré TTL du type 7485, le nombre de signaux à transmettre à l'opérateur 3 peut être deux ou trois fois plus élevé.

L'opérateur 3 est généralement un opérateur réalisant l'opération logique ET sur les résultats de comparaisons successives effectuées par l'opérateur 2. L'opérateur 3 reçoit ces résultats par liaison 10 et, par la liaison 11, le contenu du registre de travail 4. Le résultat de la deuxième opération, ou combinaison, est transmise au registre 4 par la liaison 12.

La figure 3 représente le schéma correspondant à ce cas, réalisable par exemple à l'aide de deux circuits intégrés TTL du type 7408, et comportant 8 éléments ET référencés 3.1 à 3.8. D'autres fonctions de combinaison peuvent être réalisées à l'aide de fonctions logiques ou arithmétiques différentes, par exemple OU, implication, ou comptage. Par contre, pour les opérations de rapprochement les plus complexes comme les recherches d'encadrement ou d'ordre d'occurrence, les rôles respectifs du premier opérateur 2, du deuxième opérateur 3 et du registre de travail 4 deviennent difficiles à isoler, et peut être préférable d'utiliser un des autres modes de réalisation décrits plus loin.

Le registre de travail 4 reçoit, par la liaison 12, le résultat de l'opération de combinaison effectuée par l'opérateur 3. Son contenu est transmis à cet opérateur et à un organe d'utilisation 13 par la liaison 11. Le registre de travail 4 comporte au moins 8

bascules 4.1 à 4.8 selon la figure 4 et peut être réalisé par exemple à l'aide de deux circuits intégrés TTL du type 7475 ou de quatre circuits TTL de type 7474.

Le registre-compteur 5 a pour fonction d'adresser séquentiellement la mémoire 1 de façon à ce qu'elle fournisse successivement au premier opérateur 2 le contenu de ses cases par la liaison 6. Dans ce premier cas, il s'agira donc d'un compteur à 16 états, composé de 4 bascules binaires, par exemple à l'aide d'un circuit intégré TTL de type 7493. Les adresses ainsi définies sont transmises à la mémoire 1 par la liaison 14.

Dans le cas d'un traitement bit par bit, la mémoire 1 devra comporter de 96 à 128 cases selon le code utilisé, chaque case ayant une capacité de 8 bits. Ce sera également la largeur de la liaison 6 et l'opérateur 2 devra comporter huit circuits de comparaison élémentaires entre le bit arrivant de la source par la liaison 8 et les bits provenant de la mémoire par la liaison 6.

Deux exemples de réalisation sont représentés sur les figures 5 et 6 pour la comparaison simple égal/différent, le premier utilisant par exemple des circuits intégrés multiplexeurs à quatre voies TTL de types 74155 ou 9309 et des inverseurs TTL de type 7404 et le second deux circuits intégrés OU exclusif TTL de type 7486 et un inverseur.

La liaison 10 de sortie de l'opérateur 2 a les mêmes caractéristiques que dans le premier cas et il en est de même pour l'opérateur 3 et le registre 4, tout au moins dans de cas le plus simple où la deuxième fonction est indépendante de l'élément d'information auquel est appliquée la première fonction. Les circuits des figures 3 et 4 sont alors utilisables.

Le registre-compteur 5 sera dans ce cas un registre à 128 états et comportera 6 bascules binaires.

Vu le caractère strictement séquentiel des opérations à exécuter, la commande de celles-ci pourra être effectuée par un simple signal de référence de temps issu de la source 7 par la liaison 15. Ce signal commandera la copie dans le registre 4 du résultat provisoire issu de l'opérateur 3 par la liaison 12 et la progression simultanée du registre-compteur 5. Dans le cas où la source 7 ne fournit pas un signal de référence de temps indépendant, un circuit séparateur convenable sera utilisé pour le reconstituer à partir du signal composite issu de la source.

L'initialisation du circuit comportera deux opérations distinctes. La première est le chargement de la mémoire 1 avec les données de référence. Cela est effectué par un opérateur d'écriture représenté schématiquement en 16 sur la figure 1, avec ses liaisons de données 17 et de commande des adresses 18. L'adressage de la mémoire 1 pendant le chargement dépend beaucoup des propriétés particulières de ladite mémoire et de la forme sous laquelle sont fournies les données de référence, mais il sera en général séquentiel, comme pendant le traitement, et pourra donc être effectué par le registre-compteur 5. L'opération de chargement sera grandement simplifiée s'il est possible d'accéder à la mémoire 1 en écriture autrement que case par case, par exemple caractère par caractère dans le premier mode de fonctionnement, ou par colonne dans le second. Cela impose une structure particulière de ladite mémoire qui peut être incompatible avec certaines technologies.

La deuxième partie de l'initialisation consistera à mettre en état initial le registre de travail 4 et le registre-compteur 5. L'état initial du registre de travail 4 et la registre-compteur 5. L'état initial du registre de travail 4 sera de préférence celui pour lequel l'opérateur 3 devient transparent à la sortie de l'opérateur 2 de façon à copier dans le registre 4 la sortie de l'opérateur 2. Dans le cas où cela n'est pas possible ou trop difficile à réaliser, vu par exemple l'absence d'élément neutre pour la deuxième fonction choisie pour l'exécution du rapprochement, on devra tenir compte de l'influence de l'état initial du registre 4 lors de l'interprétation du résultat du rapprochement, état final du même registre, par l'organe d'utilisation et de décision 13.

On peut également utiliser l'initialisation du registre de travail 4 pour limiter l'application du rapprochement à une partie des entités logiques de référence contenue dans la mémoire 1. C'est le cas notamment avec l'opérateur représenté par la figure 3. Un état initial 0 ou faux dans une des bascules du registre de travail 4 est équivalent à l'inhibition du rapprochement avec l'entité logique de référence correspondante.

Le registre-compteur 5 sera initialisé dans l'état qui adresse la première case de la mémoire 1. Cette initialisation devra être effectuée au début de chaque entité logique reçue et sera commandée soit par un comptage dans le cas d'entités de taille fixe, soit par la détection de caractères particuliers dans le cas d'entités de taille variable. Ces tests seront effectués par un organe particulier de commande représenté en 19 sur la figure 1 et qui reçoit les signaux d'information et de temps de la source 7 par les liaisons 8 et 15 et fournit par les liaisons 20 et 21 des signaux d'initialisation des registres 4 et 5. Cet organe de commande reçoit aussi du registre-compteur 5 un signal de fin de cycle par la liaison 22 et fournit également un signal de fin d'opération à l'organe d'utilisation et de décision 13 par la liaison 23, en vue de provoquer le prélèvement du résultat qui est alors contenu dans le registre de travail 4 et son exploitation. Comme il est représenté par la figure 1, la liaison de sortie 11 peut aussi aboutir à l'organe de commande 19, pour lui permettre de tenir compte du résultat d'un rapprochement pour l'initialisation du rapprochement suivant.

Dans une variante de réalisation, représentée sur la figure 7, les fonctions effectuées dans le premier mode de réalisation par l'opérateur 3 et par le registre de travail 4 sont assurées par un opérateur unique 25. Celui-ci n'a alors qu'une entrée de données, la liaison 10, une sortie la liaison 11, et deux liaisons de commande 15 et 20, toutes ces liaisons ayant le même rôle que dans le premier mode de réalisation. Ladite variante conduit souvent à des circuits plus économiques que celle de la figure 1, et elle est particulièrement bien adaptée à tous les cas où la deuxième fonction est une opération arithmétique ou de comptage. C'est ce type d'opération qu'on utilise généralement dans les opérations de rapprochement complexes, comme les évaluations de distance de Hamming, par exemple.

La figure 8 représente un exemple de réalisation selon cette variante d'une comparaison simple égal/différent, en mode bit par bit. Le circuit représenté comporte un étage de l'opérateur 2, constitué par un élément OU EXCLUSIF 26, et un étage de l'opérateur 25 constitué par trois éléments NI 27 à 29. Les éléments NI 28 et 29 forment une bascule SET-RESET, l'élément NI 27 est relié à l'entrée RESET de cette bascule.

A l'initialisation, le signal provenant de l'opérateur de commande 19 par la liaison 20 arme l'ensemble des bascules de l'opérateur 25. Chaque unité d'information provenant de la source 7 par la liaison 8 est ensuite comparée en parallèle avec celles issues de la mémoire 1 par la liaison 6, à l'aide des éléments OU EXCLUSIF 26. Dans le cas d'une discordance entre le signal 8 et un des signaux 6, le signal de sortie de l'élément 26 correspondant, transmis par la liaison 10 à l'élément NI 27, permet au signal de commande 15 de désarmer la bascule SET-RESET. Ainsi, à la fin de l'opération de rapprochement, une bascule restée armée indique qu'il n'y a aucune discordance entre l'entité logique de référence correspondante et l'entité logique reçue de la source, c'est-à-dire qu'elles étaient identiques. Le contenu des bascules SET-RESET de l'opérateur 25, accessible via la liaison 11, donne donc bien un critère d'identification de l'entité logique reçue.

En associant un ensemble de 8 circuits de ce type avec une mémoire à lecture seule de 8 cases de 8 bits et un circuit de commande simplifié, on réalise simplement un circuit d'identification de caractères utilisable pour l'identification des caractères spéciaux et en particulier du caractère de séparation des entités logiques de taille variable. L'organe ainsi défini peut

donc servir d'organe de commande aux opérateurs d'un système de rapprochement destiné à traiter des entités logiques de taille variable, c'est-à-dire à constituer l'essentiel de l'organe représenté en 19 sur la figure 1. La mémoire indiquée plus haut pour cet organe n'est donnée qu'à titre d'exemple, le nombre de cases et leur taille pouvant varier selon la taille et le nombre des caractères spéciaux. De même, une mémoire à lecture-écriture peut être employée dans le cas où on doit traiter des données représentées dans des codes différents, ou contenant des jeux de caractères spéciaux différents.

Selon une seconde variante représentée par la figure 9, les fonctions des opérateurs 2 et 3 sont assurées par un opérateur unique 30 recevant des données par les liaisons 6, 8 et 11, et transmettant le résultat au registre 4 par la liaison 12. La figure 10 donne un exemple de réalisation de l'opérateur 30 dans le cas d'une comparaison binaire du type plus grand, plus petit, plus grand ou égal, ou plus petit ou égal. Il peut être appliqué à des données de nature quelconque par exemple numériques, en traitement bit par bit, poids faibles en tête. Pour des données non numériques, cela peut constituer un opérateur de tri. L'opérateur 30 comprend alors un inverseur 31 et 8 éléments logiques triple ET-OU, 32.1 à 32.8 qui évaluent, chacun, séquentiellement le report de la soustraction binaire effectuée entre l'entité logique reçue et une des entités logiques de référence. Le choix entre les quatre tests possibles est défini par le contenu initial du registre de travail 4 et par l'interprétation du résultat, contenu final du même registre. Cette spécification peut être différente pour les 8 entités logiques de référence, ce qui permet d'effectuer simultanément des comparaisons de types différents, et réaliser ainsi directement des recherches d'encadrements ou de limites.

Selon une troisième variante représentée sur la figure 11, on combine les fonctions des deux opérateurs 2 et 3 et du registre de travail 4 en un opérateur unique 35.

Bien entendu, de nombreuses autres variantes sont possibles. De même, tous les exemples ont été décrits pour un ensemble de 8 entités logiques de référence comportant chacune au maximum 16 caractères ou 128 bits.

Selon encore une autre variante représentée sur la figure 12, le circuit de rapprochement comporte plusieurs modules de mémoires tels que 1a-n contenant des ensembles différents de données de référence. Tous ces modules sont adressés en parallèle par le registre-compteur 5 par la liaison 14. La sélection du module utilisé à un instant donné est définie par l'organe de commande 19 par la liaison 40.

Selon encore une autre variante représentée sur la figure 13, les différents ensembles de données de référence sont contenus dans différentes parties d'une mémoire commune 41. L'adressage de ladite mémoire est alors effectué par la combinaison du contenu du registre-compteur 5 par la liaison 14 et d'un complément d'adresse fourni par l'organe de commande 19 par la liaison 42.

Selon encore une autre variante, le ou les opérateurs 2, 3, 25, 30 ou 35 sont constitués de plusieurs opérateurs différents recevant les mêmes données et dont l'activité est commandée par l'organe de commande 19.

Par exemple, tel que représenté sur la figure 14, un opérateur 2 est remplacé par un ensemble d'opérateurs indépendants 2.a, 2.b, à 2.n, réalisant autant de premières fonctions différentes. Ces opérateurs reçoivent leurs données d'entrée en parallèle de la mémoire 1 et de la source 7 par les liaisons 6 et 8 et leurs sorties sont regroupées en une sortie unique 10. Le choix de l'opérateur 2 actif à un instant donné est commandé par l'organe de commande 19 par une liaison 43.

Selon encore une autre variante, les opérateurs 2 et 3 ou les opérateurs combinés 25 ou 30 sont constitués à l'aide de circuits logiques multifonctions tels que les circuits intégrés TTL de type 74181. Cette disposition permet de réaliser de façon compacte

un système capable d'effectuer des rapprochements de natures diverses.

Selon un deuxième mode de réalisation représenté par la figure 15, le tableau des données de référence est contenu dans une mémoire 50. Cette mémoire est adressée par une combinaison de la sortie 14 du registre-compteur 5 et de l'élément d'information reçu de la source 7 par la liaison 8. Cette combinaison sélectionne une case de la mémoire 50, par l'intermédiaire du circuit de décodage 50a, case qui contient le résultat de la comparaison selon la première fonction de l'élément d'information reçu avec les éléments d'information de même rang des entités logiques de référence.

Le contenu de ladite case issu de la mémoire 50 par la liaison 51 est combiné selon la seconde fonction par l'opérateur 3 avec le contenu du registre de travail 4 disponible sur la liaison 11. Le résultat de cette combinaison est inscrit dans le registre de travail 4 par la liaison 12 et le signal de commande issu de la source 7 par la liaison 15. Le même signal fait progresser en même temps le registre-compteur 5 en vue de préparer le traitement de l'élément d'information suivant.

L'initialisation de la mémoire 50 est effectuée à l'aide de l'opérateur 52 qui permet de comparer selon la première fonction les éléments d'information successifs des entités logiques de référence présentées sur la liaison 54 avec les différentes configurations présentées par le registre-compteur auxiliaire 55 sur la liaison 56. La liaison 57 permet de faire progresser le registre-compteur 5 d'un pas à chaque cycle du registre-compteur 55. L'adressage de la mémoire 50 est effectuée par les liaisons 14 et 56. Enfin, une liaison 58 permet de procéder à un chargement direct de la mémoire 50, en cas par exemple d'opération répétitive, à l'aide de données déjà préparées.

Ce mode de réalisation peut comporter les mêmes variantes de réalisation que ceux exposés à propos du premier, à l'exception de celles représentées par les figures 9 et 11.

L'organe d'utilisation et de décision 13 pourra comporter plusieurs opérateurs selon le type de résultat recherché. Un premier opérateur simple est une mémoire tampon où seront enregistrés les résultats successifs de rapprochement sur les entités logiques d'une séquence ou d'une suite de séquence, ce stockage est commandé par l'organe 9 via la liaison 23 (figure 1).

Un deuxième opérateur important est un registre de récapitulation. Comme représenté par la figure 16, il s'agit d'un registre comportant ici 8 bascules 100.1 à 100.8 associé à un opérateur logique OU formé des 8 éléments 101.1 à 101.8.

A cet opérateur de récapitulation peut être associé un certain nombre d'opérateurs de décision fournissant un résultat binaire et aptes ainsi à fournir des signaux de commande. Le premier est un circuit OU à 8 entrées 102 qui permet de décider si au moins un des résultats a été obtenu. Le second 103 est un circuit ET e même type qui indique par sa sortie si tous les résultats ont été obtenus, par exemple si toutes les entités logiques de référence ont été reconnues.

Le troisième est composé d'un comparateur 104 et d'un registre 105. Il permet de comparer la récapitulation des résultats contenus dans le registre 100 avec une certaine configuration affichée dans le registre 105.

Enfin, les sorties 11 et 106 respectivement du registre de travail 4 et du registre de récapitulation 100 peuvent être reliées directement ou à l'aide d'un registre intermédiaire à l'entrée d'un autre système de rapprochement qui permettra de comparer les résultats de rapprochements individuels ou récapitulés à un ensemble de résultats types contenus dans une deuxième mémoire analogue à 1.

Il est également possible d'appliquer ce système à l'identification d'entités logiques de taille variable par exemple de genre de celles servant de délimiteurs pour des séquences ou des suites de séquences. Le système comporte alors de préférence trois

sous-systèmes de rapprochements tels que représentés par la figure 17, destinés le premier 110 à l'identification des caractères spéciaux, le second 111 à l'identification des entités logiques «délimitateurs», et le troisième 112 au rapprochement principal.

Les données à rapprocher issues de la source 7 sont reçues en parallèle par les trois sous-systèmes qui fonctionnent ainsi simultanément en synchronisme avec le débit de ladite source mais selon un cycle différent. Une mémoire tampon 113 permet le stockage temporaire d'une séquence d'entités logiques.

L'initialisation de chacun de ces sous-systèmes est alors commandée par un organe de traitement 114, selon les résultats obtenus par le ou les précédents sous-systèmes.

Les résultats du rapprochement principal sont transmis à l'organe de traitement 114 par le sous-système 112, via la liaison 115, accompagné des paramètres d'identification issus notamment du sous-système 111, et éventuellement des séquences stockées dans la mémoire tampon 113.

La figure 18 représente l'organisation logique d'une mémoire bien adaptée à la réalisation des systèmes de rapprochement. Cette mémoire est formée de  $h$  modules identiques 200.1 à 200.h comprenant chacun  $n$  cases de  $p$  bits.

Les entrées d'adresse et de données de ces modules sont reliées respectivement aux liaisons parallèles 201 et 202. Les sorties de données de ces modules sont des liaisons parallèles indépendantes 203.1 à 203.h. Dans certains systèmes, la mémoire comprendra également des multiplexeurs 204.1 à 204.h, dont les entrées sont reliées aux liaisons de sortie des modules 203.1 à 203.h, et dont les sorties constituent la liaison parallèle 205.

La commande en écriture des modules de mémoire 200.1 à 200.h est effectuée séparément pour chaque module par des liaisons simples 206.1 à  $h$ . Ces modules sont commandés collectivement en lecture, si cela est technologiquement nécessaire, par la liaison 207. Les multiplexeurs de sortie 204.1 à  $h$  sont commandés, quand ils existent, collectivement par la liaison parallèle 208.

Lors de l'initialisation de système de rapprochement, les entités de référence seront enregistrées successivement dans les modules de mémoire 200.1 à  $h$ , à raison d'une entité par module. Les éléments d'information de  $p$  bits constituant les entités logiques de référence seront présentées successivement sur la liaison parallèle 202, et enregistrées chacune dans la case désignée par l'adresse présentée simultanément sur la liaison parallèle 201. L'opération d'écriture sera commandée par un signal appliqué sur la liaison 206 aboutissant au module de mémoire 200 correspondant à l'entité de référence en cours d'écriture.

Les  $h$  éléments d'information de  $p$  bits apparaissant simultanément sur les sortiers parallèles 203.1 à  $h$  peuvent être directement utilisés pour l'exécution d'une opération de rapprochement effectuée sur des éléments d'information de  $p$  bits issus de la source 7, des caractères par exemple, telle que décrite aux pages 7, 9 et 24 et illustrée par les figures 1 et 2. Ces sorties pourront être également utilisées pour le transfert d'une partie du contenu d'un ou plusieurs modules, dans tous les cas où celle-ci est utilisée comme résultat.

Les multiplexeurs 203.1 à  $h$  permettent d'obtenir sur la liaison parallèle 205 la séquence dans l'ordre convenable des unités d'information, ou bits, constituant les entités logiques de référence, séquence nécessaire à l'exécution bit par bit d'une opération de rapprochement. Dans ce cas, l'ensemble des liaisons parallèles d'adresse 201 et de commande 208 constitue la liaison 14 des figures 1, 7, 9, 11, 12 et 13 qui constitue la sortie du registre-compteur 5, la liaison 208 de commande des multiplexeurs 204 constituant la sortie des étages dits de «faible poids» dudit registre 5.

Une telle organisation logique de la mémoire contenant la table des entités logiques de référence permettra notamment

son changement rapide à partir de n'importe quelle source capable de fournir ces entités successivement, caractère par caractère, ou plus généralement, par éléments de  $p$  bits. Cela est particulièrement important dans le cas d'un système tel que celui représenté schématiquement par la figure 17, car les entités de référence «délimitateurs» contenues dans la mémoire du sous-système 111 seront issues en général de la source 7 avec un débit semblable à celui des données à rapprocher.

Ladite organisation logique peut bien entendu être réalisée selon des modes dépendant des composants disponibles. Chaque module de mémoire peut être par exemple formé de plusieurs circuits intégrés semblables, adressés et commandés en parallèle. Un module formé de deux circuits intégrés mémoire 16 x 4 bits TTL 7489 permettra de stocker des entités logiques ayant au maximum 16 caractères. Le chargement de ce module pourra se faire soit par octet, soit par élément de 4 bits. La lecture du contenu d'un tel module pourra être effectué symétriquement par octet ou par élément de 4 bits, ou encore bit par bit à l'aide d'un multiplexeur à 4 ou 8 entrées, ou par toute combinaison de deux de ces modes.

Par contre, tout mode de réalisation utilisant un composant à grande capacité pour contenir plusieurs modules de mémoire tels que 200.1 à 200.h entraînera une exécution plus complexe des opérations d'écriture ou/et de lecture, selon l'organisation pratique choisie. Le groupement dans une même case d'éléments d'information appartenant à plusieurs entités logiques de référence exigera lors de chaque cycle d'enregistrement, la lecture préalable des éléments déjà enregistrés et, par conséquent, leur stockage temporaire dans un registre.

La répartition dans plusieurs cases d'adresses différentes d'éléments d'information de même rang appartenant à différentes entités logiques de référence imposera une lecture successive de ces éléments, nécessaires à l'exécution d'un même cycle de l'opération de rapprochement, et, par conséquent, une multiplication par le même facteur du temps d'exécution de l'opération.

Néanmoins, une telle organisation peut être parfaitement justifiée par l'économie permise par l'utilisation d'un seul ou de quelques circuits intégrés à grande capacité dans tous les cas où le débit de la source d'information le permet.

Le système représenté sur la figure 19 comporte trois mémoires de tables, TCS 300 contenant les caractères spéciaux, TELS 301 les entités logiques spéciales, et TELR 302 les entités logiques de référence, plus une mémoire TER 303 contenant les résultats types.

Ces mémoires sont constituées, comme représenté sur la figure 18, de modules identiques 200.1 à 200.h et d'autant de multiplexeurs 204.1 à 204.h.

La mémoire TER 303 peut également être constituée d'un certain nombre de registres à recirculation commandés en parallèle.

Les mémoires de table TCS 300, TELS 301 et TELR 302 sont adressées par un organe commun d'adressage A 304 composé de deux registres-compteurs 305 et 306 montés en cascade, le registre-compteur 306 étant du type non cyclique, c'est-à-dire qu'il ne revient en état initial que sous l'action de la commande de remise à zéro 327.

La sortie 307 du registre-compteur 305 correspond aux positions basses de l'organe d'adressage A 304, et est commune à l'ensemble des trois mémoires de table 300 à 302. La sortie 308 du registre-compteur 306, qui correspond au nombre maximal de caractères des entités logiques spéciales et de référence, commande seulement les mémoires de table TELS 301 et TELR 302.

Une liaison parallèle 309 permet de remplir ces mémoires de table, par exemple caractère par caractère.

Une source 310 fournit des données en mode série par la liaison 311, et un signal de référence temporelle par la liaison 312.

La comparaison entre ces données et les sorties 313, 314 et 315 des mémoires de table TCS 300, TELS 301 et TELR 302 est effectuée par trois opérateurs 316, 317 et 318. Les deux premiers 316 et 317 correspondent à la fonction égal-différent et peuvent être constitués, par exemple d'un certain nombre de circuits tels que celui représenté sur la figure 8, ou tout circuit logiquement équivalent utilisant des commandes par changement d'état. L'opérateur 318 peut avoir une structure identique, ou être constitué de circuits tels que ceux représentés aux figures 3, 5, 6 et 10 ou, encore, être constitué, comme il est dit ci-dessus, de circuits intégrés à fonctions multiples permettant une grande variété de fonctions de rapprochement.

La sortie de l'opérateur 318 est recueillie par l'opérateur 319 qui effectue la combinaison des résultats de rapprochements élémentaires selon une troisième fonction logique telle que décrite ci-dessus, et représentée à titre d'exemple en 100 et 101 sur la figure 16.

Une sortie 320 de l'opérateur 319 est reliée à l'opérateur 321 qui a pour but de rapprocher le contenu final de l'opérateur 319 avec le contenu de la mémoire de table TER 303 disponible sur la liaison 322. La structure de l'opérateur 321 est analogue à celle de l'opérateur 318.

La sortie 325 de l'opérateur de rapprochement 316 est reliée à un opérateur logique d'initialisation 326 qui élabore, à partir des signaux de la liaison 325, un signal sur la liaison 324 de commande de l'organe de décision 335, un signal sur la liaison 327 de remise à zéro de l'organe d'adressage A 304, un signal sur la liaison 328 de remise à zéro des registres de travail des opérateurs 316 à 318, et un signal sur la liaison 329 de transfert du contenu de l'opérateur 317 par sa liaison de sortie 330. Ce transfert est effectué dans un registre 331 dont les sorties 332, 333 et 334 commandent respectivement: le transfert du résultat de l'opérateur 318 vers l'opérateur 319; l'initialisation de cet opérateur 319 et de l'opérateur 321; et certaines fonctions de l'organe de décision 335.

Cet organe de décision 335 reçoit par la liaison 323 le contenu final de l'opérateur 319 et comporte, comme représenté sur la figure 16, un certain nombre de circuits qui effectuent sur le résultat du rapprochement contenu dans l'opérateur 319 des tests logiques simples du type: résultat total, résultat partiel, résultat majoritaire, résultat dans un ordre donné, etc... L'ensemble des résultats de ces tests peut être stocké dans la mémoire-tampon 336 par la liaison 337 et la liaison de commande 338.

L'organe de décision 335 fournit également des signaux de commande, d'initialisation et d'écriture sur la liaison 339 à la mémoire-tampon 340 qui reçoit, par les liaisons 311 et 312, les données et la référence temporelle issues de la source de données 310.

La mémoire-tampon 336 est également reliée par la liaison 341 à la sortie de l'opérateur 321 de comparaison avec les résultats-types.

Enfin, la sortie 342 permet à l'organe utilisateur de consulter le contenu de la mémoire-tampon 336, c'est-à-dire les résultats de tests et de comparaison avec les résultats-types. De même, la sortie 343 de la mémoire-tampon 340 permet d'exploiter les séquences de données qui y ont été stockées.

La description ci-dessus correspond essentiellement à un système minimal travaillant bit par bit.

Le même schéma représente également l'organisation d'un système fonctionnant par exemple octet par octet. Dans ce cas, les mémoires de table 300 à 302 ne comporteront plus de multiplexeurs tels que ceux référencés 204.1 à 204.h dans la figure 18, et des opérateurs 316 à 318 comporteront en tête un premier opérateur parallèle tel que celui représenté sur la figure 2.

Par ailleurs, l'opérateur d'initialisation 326 et le registre 331 peuvent comporter un plus grand nombre de sorties correspon-

dant à des fonctions différentes de celles représentées à la figure 19, telles que celles énoncées ci-dessus.

Ceci suppose une capacité correspondante des mémoires de tables 300 et 301. Certaines sorties supplémentaires de l'organe 326 peuvent remplacer tout ou partie de celles représentées pour le registre 331 ou, encore, être combinées avec celles-ci par un circuit logique OU. Cette substitution ou, plus généralement, la répartition des signaux de commande issus des opérateurs 326 et 331 peut être commandée par l'organe utilisateur de façon à adapter la logique d'initialisation et d'enchaînement de système de rapprochement à la structure particulière d'une base de données et aux besoins d'un utilisateur.

De même, des sorties supplémentaires de l'organe de décision 335 peuvent fournir des signaux d'interruption, de requête ou d'état à un organe utilisateur, de type ordinateur par exemple, et des signaux de commande à l'organe source de données en vue de son fonctionnement séquentiel normal, ou d'un fonctionnement particulier comme, par exemple, un signal de commande d'écriture à l'électronique de couplage d'un disque magnétique, sur la reconnaissance de l'entête caractéristique d'un enregistrement particulier.

Dans l'exemple de réalisation décrit ci-dessus, pour pouvoir exécuter des reprises en cas d'erreur, il sera nécessaire de réaliser la mémoire-tampon 340 sous la forme d'une mémoire à accès aléatoire accompagnée d'un registre d'adresse et ce registre d'adresse sera muni d'une doublure à liaisons croisées.

Comme représenté à la figure 20, la mémoire-tampon 340 comprend alors un module de mémoire à lecture-écriture 350, un registre d'adresse 351 et un registre-doublure 352. Les sorties respectives 353 et 354 de ces deux registres sont reliées à une entrée de l'autre registre permettant ainsi la copie réciproque de leur contenu de façon non destructive sous l'action des commandes 355 et 356. De plus, le registre d'adresse 351 possède une commande d'initialisation 357 et une commande de progression 358. En fonctionnement normal, la commande de progression 358 permet un enregistrement séquentiel dans le module de mémoire 350 des données arrivant par la liaison 360 ou leur lecture séquentielle par la liaison 361, selon l'état de la commande de lecture-écriture 359.

La mémoire-tampon de résultat 336 peut également avoir une structure semblable, à l'exception de la liaison d'entrée 360 qui est alors munie d'un multiplexeur à deux entrées 337 et 341.

Lors de la lecture d'une base de données ou de la réception de messages, la commande 355 permet de stocker dans le registre-doublure 352 un pointeur de reprise à la fin de tout enregistrement ou message lu ou reçu correctement en détruisant en même temps le pointeur de reprise précédent.

L'enregistrement ou le message suivant, reçu de la source par la liaison 311, est inscrit dans le module-mémoire 350 par la liaison 360 alors reliée à la liaison 311, avec progression des adresses par la liaison 358 alors reliée à la liaison 312.

En cas d'erreur, l'exécution de la reprise sera effectuée par action sur la commande 356 qui provoque, dans les deux mémoires-tampons 336 et 340, la copie du registre-doublure 352 dans le registre d'adresse 351.

En l'absence d'erreur, la commande 355 remplace le pointeur de reprise à la fin des données qui viennent d'être stockées dans les deux mémoires-tampons 336 et 340.

Une telle reprise peut être exécutée à un moment quelconque, et un nombre quelconque de fois selon les méthodes habituelles utilisées dans la gestion des fichiers et des transmissions de données.

Cette disposition des mémoires-tampons a également l'avantage de permettre la suspension du rapprochement en cas de débordement de l'une des mémoires-tampons en vue de transférer vers l'organe utilisateur la partie de leur contenu correspondant à des enregistrements ou messages lus ou reçus et

traités complètement, et délimitée par le pointeur stocké dans le registre-tampon 352, tout en permettant ensuite la reprise du traitement avec répétition de l'enregistrement ou message partiellement traité.

L'information stockée dans des modules-mémoires 350 est alors récupérée par l'organe utilisateur par les liaisons 361 (figure 20) appelées 342-342 dans la figure 19, et les liaisons de commande 358 (figure 20).

Ce dispositif permet également d'effectuer de façon simple et automatique des modifications sur le contenu d'un enregistrement d'une base de données. Lors de la lecture de cet enregistrement, la position de l'article ou du sous-article à modifier est repérée par les dispositifs de rapprochement et conservée dans le registre-doublure 352 sous l'action de la commande 355 alors issue de l'organe de décision 335, lors de l'identification d'un caractère spécial ou d'une entité logique spéciale particulière. La lecture est poursuivie jusqu'à la fin de l'enregistrement avec stockage de la totalité de celui-ci dans le module-mémoire 350 de la mémoire-tampon 340 par l'intermédiaire des liaisons 311 et 360. La copie du pointeur dans le registre-doublure 352 dans le registre d'adresse 351 par la commande 356 permet alors d'effectuer la modification sur la séquence ainsi repérée. Le registre d'adresse 351 est ensuite ré-initialisé par action sur la commande 357 et le contenu modifié du double de mémoire 350 est ré-enregistré sur le support de la base de données, par exemple au même endroit si l'étendue de la modification le permet. Cette ré-écriture est effectuée par une lecture séquentielle de la mémoire 350 sous l'action des commandes 358 et 359, la liaison 361 alors reliée à l'entrée de données de l'organe de stockage de base de données, disque magnétique par exemple.

Dans le cas d'une modification simple, par exemple la remise à jour d'un compte ou de tout autre paramètre numérique, les éléments de la modification, code opération et données, peuvent être contenus dans une table spéciale annexée au dispositif. Le choix d'un de ces éléments est commandé alors directement par la sortie 323 de l'opérateur 319 qui identifie l'article repéré parmi tous ceux qui peuvent être recherchés par rapprochement de façon simultanée, et dont les descriptifs sont contenus dans la table 302.

Ce dispositif permet encore d'effectuer un stockage systématique d'entités logiques soumises à rapprochement et telles que le résultat de ce rapprochement conditionne leur transfert effectif vers l'organe utilisateur. Le pointeur contenu dans le

registre-doublure 352 définit alors de début d'une telle séquence, et sa copie par la commande 356 dans le registre d'adresse 351 sous le contrôle de l'organe de décision 335 permet la ré-écriture d'une séquence ultérieure à la même place, avec destruction de la séquence ultérieure à la même place, avec destruction de la séquence non demandée.

Ce processus peut s'appliquer à des suites de séquences avec stockage total ou partiel, la décision résultant, d'une combinaison de résultats de rapprochements portant sur tout ou partie des séquences de la suite.

Un tel processus demande, en général, le stockage de plusieurs pointeurs pour le repérage d'autant d'entités logiques ou séquences d'entités logiques, à l'intérieur d'une même suite, avant qu'un traitement ou une décision puisse intervenir. Pour cela, le registre-doublure 352 est remplacé par une structure de file (en anglais «first in first out», et par abréviation FIFO) ou de pile (en anglais «last in first out», et par abréviation LIFO) permettant le stockage desdits pointeurs avec reprise et utilisation dans un ordre identique ou inverse à celui de leur création.

De même, si l'on veut bénéficier simultanément de plusieurs de ces processus de reprise, de modification et/ou de sélection a priori des séquences stockées, on utilise plusieurs registres-doublure tels que 352, commandés chacun indépendamment par les circuits de détection des événements correspondants, et qui peuvent être recopiés chacun indépendamment dans le registre d'adresse 351, en vue d'exécuter la reprise en cas d'erreur, la modification, ou la décision de non utilisation concernant le dernier enregistrement ou message.

Plus généralement, on peut remplacer, dans les mémoires-tampons 336 et/ou 340, le registre-doublure 352 par une combinaison de registre, files et piles, permettant une combinaison quelconque de processus tels que décrits ci-dessus.

Enfin, un doublement du registre de l'opérateur 319 permet une reprise en cas d'erreur ou de débordement, pour des séquences ou suites de séquences s'étendant sur plusieurs enregistrements, ou plusieurs messages. Un traitement analogue de l'organe d'adressage 304, des registres de travail des opérateurs 317 et 318 du registre 331 permettrait d'effectuer des reprises pour des sources de données où une entité logique peut se trouver à cheval sur deux enregistrements ou messages, rendant ainsi totalement indépendantes les structures logiques et physiques de la source de données, tout en permettant un traitement immédiat des données lues ou reçues.

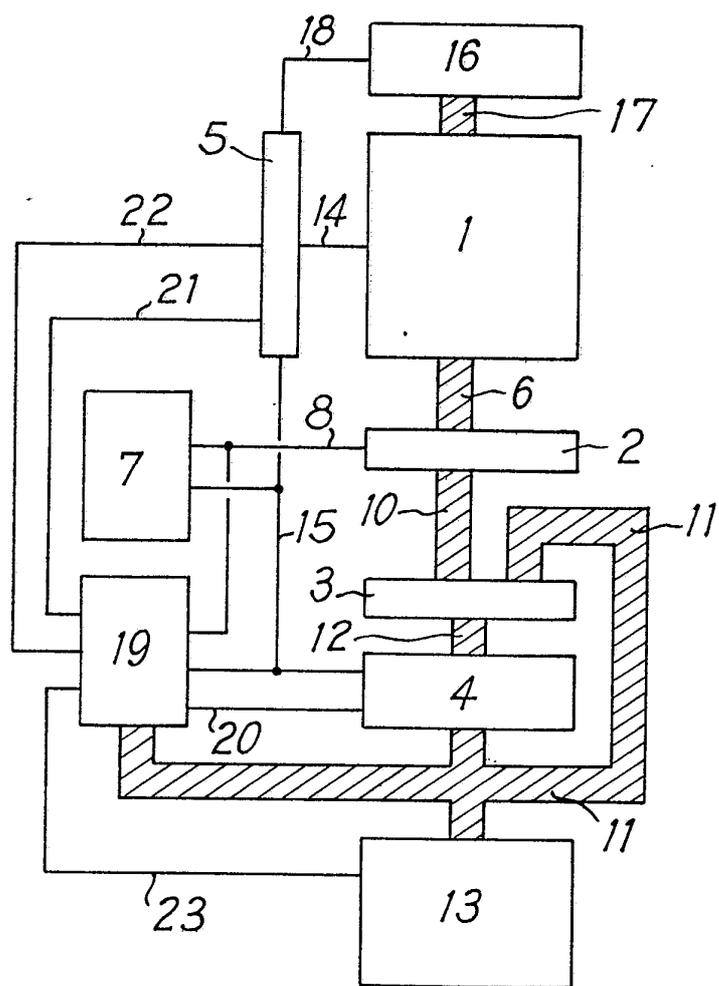


Fig. 1

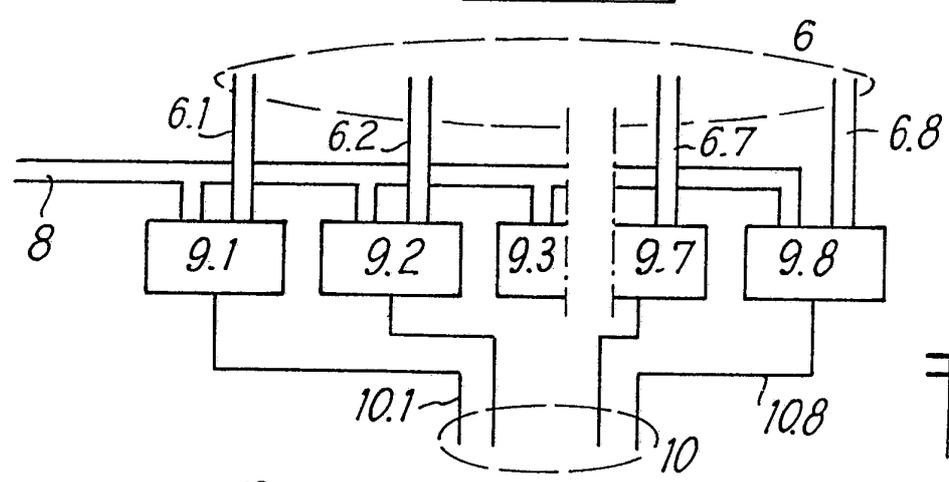


Fig. 2

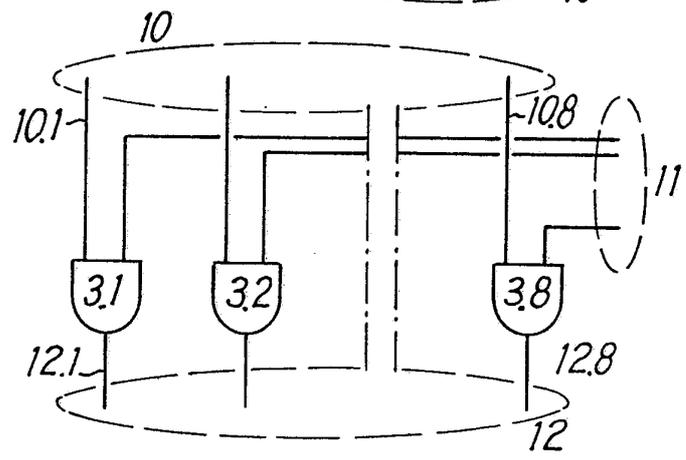


Fig. 3

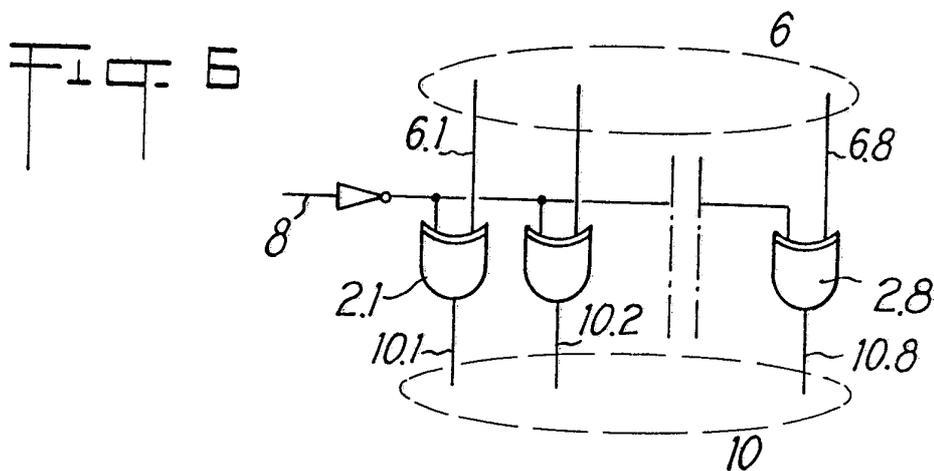
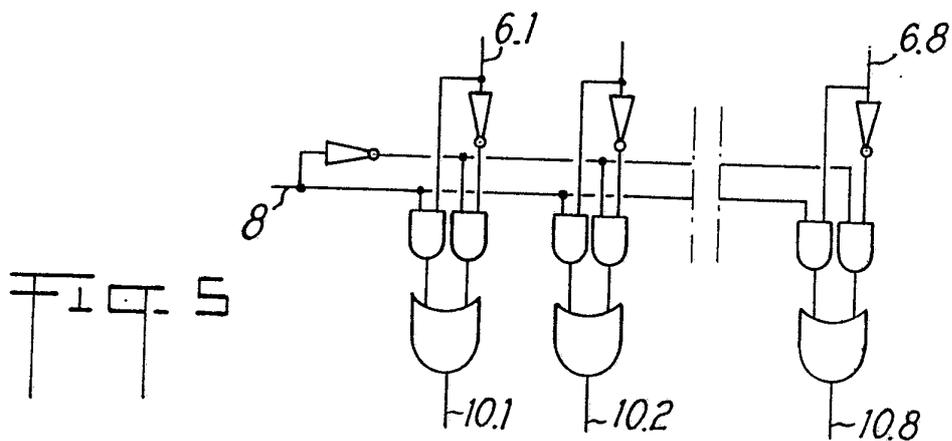
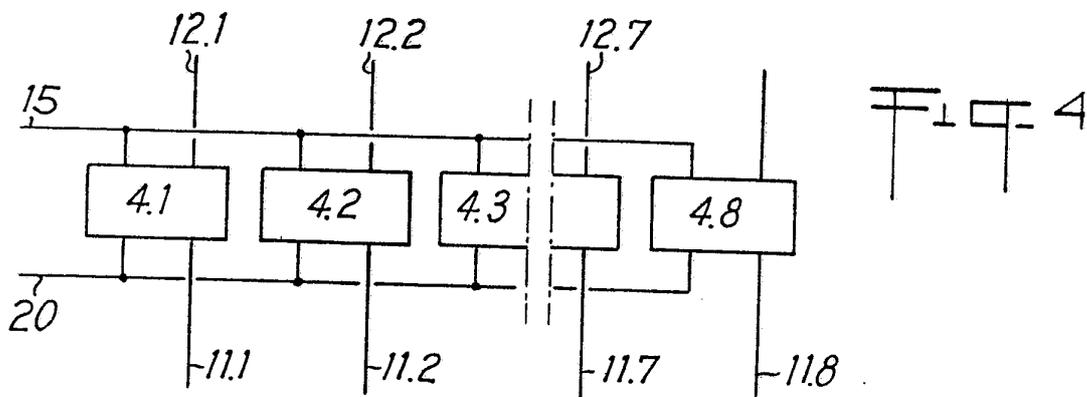


Fig. 7

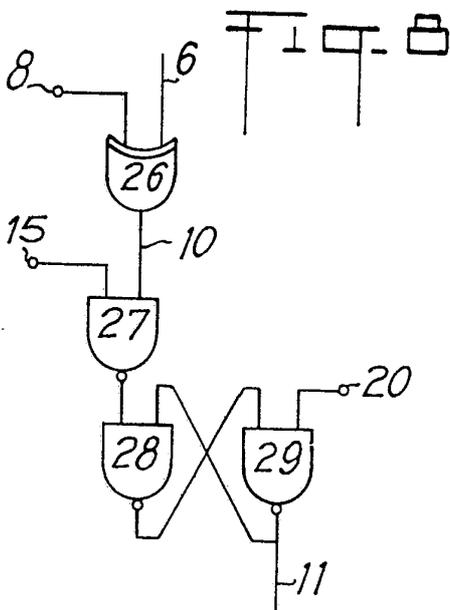
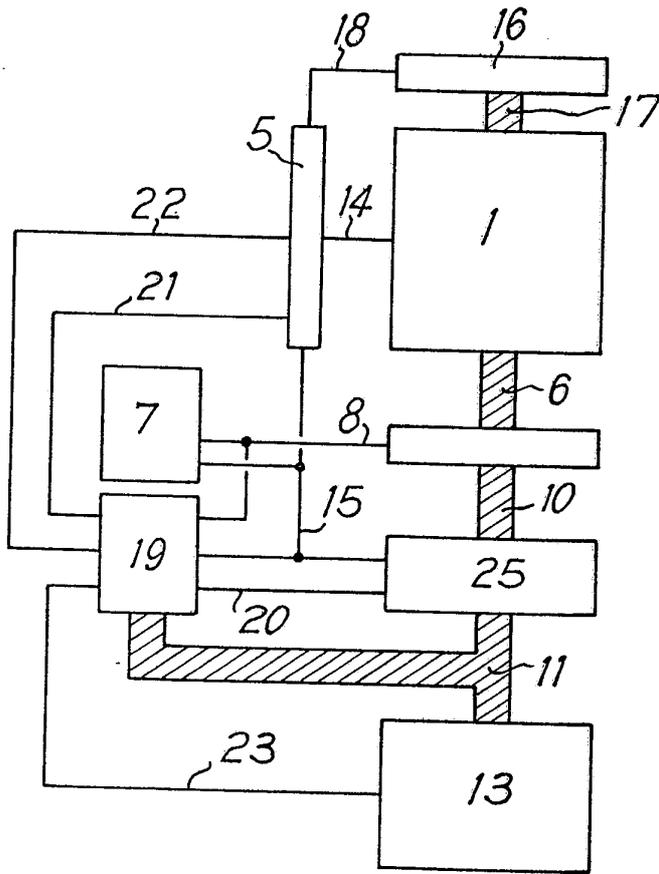
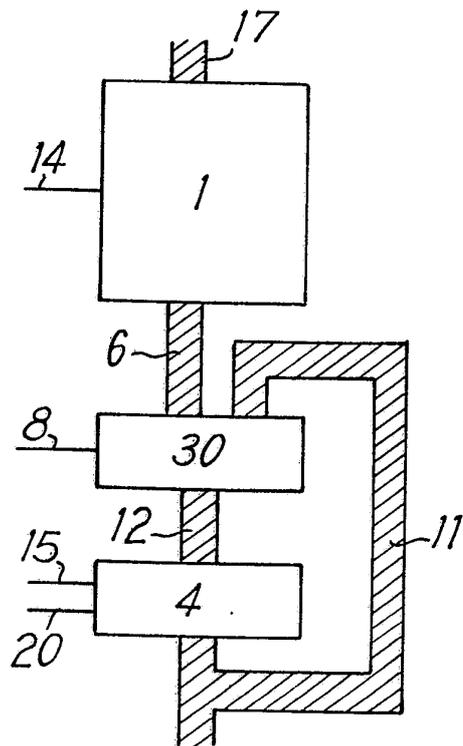


Fig. 9



ANVAR

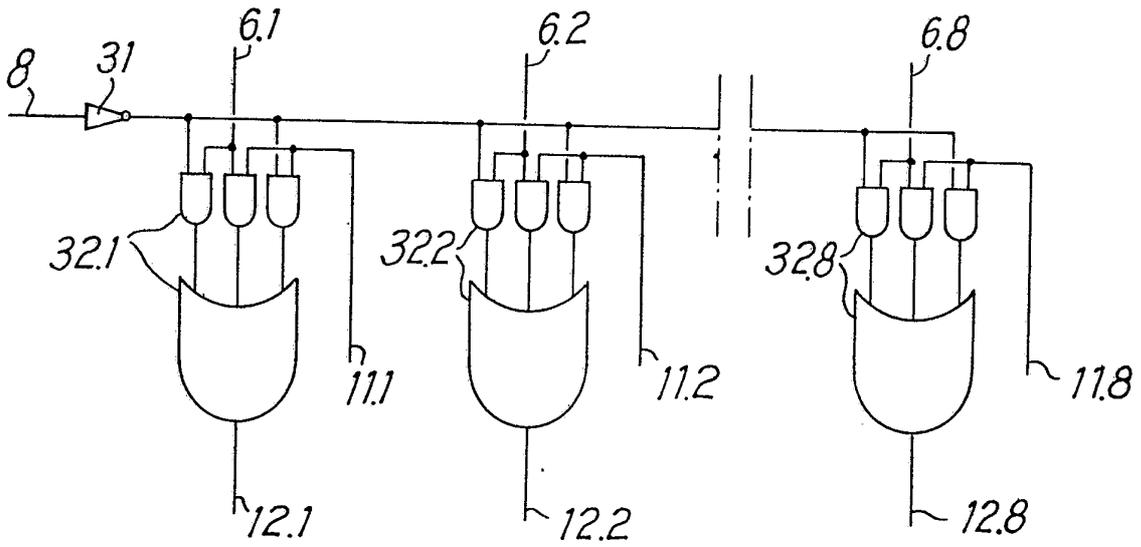


Fig. 10

Fig. 11

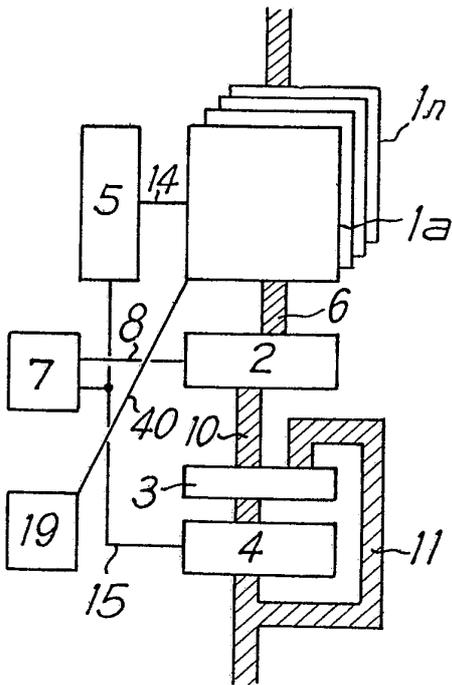
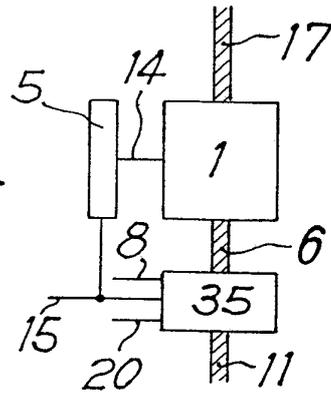


Fig. 12

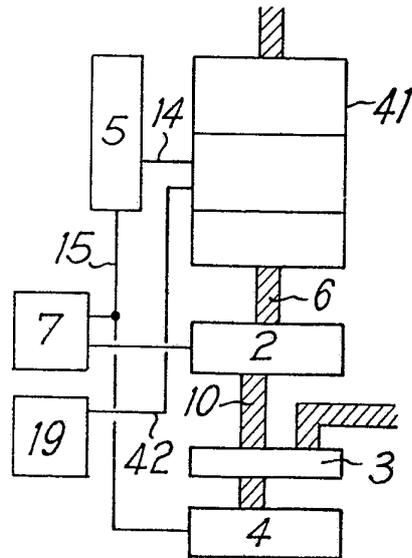


Fig. 13

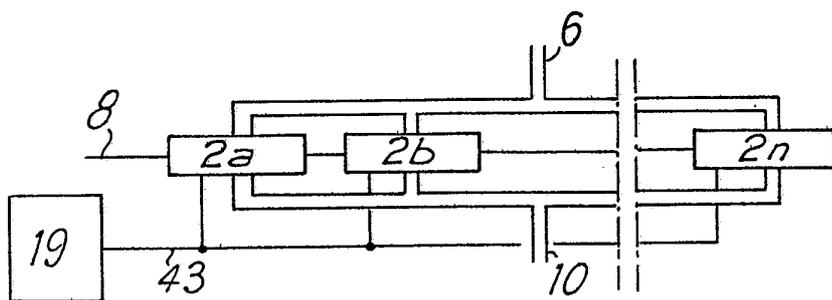


Fig. 14

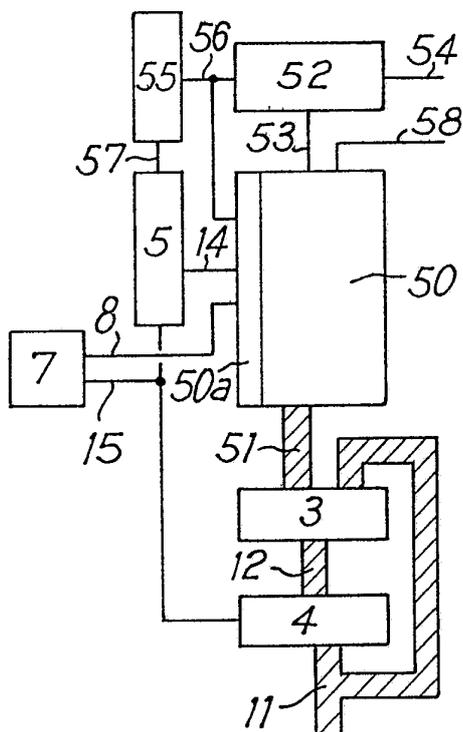


Fig. 15

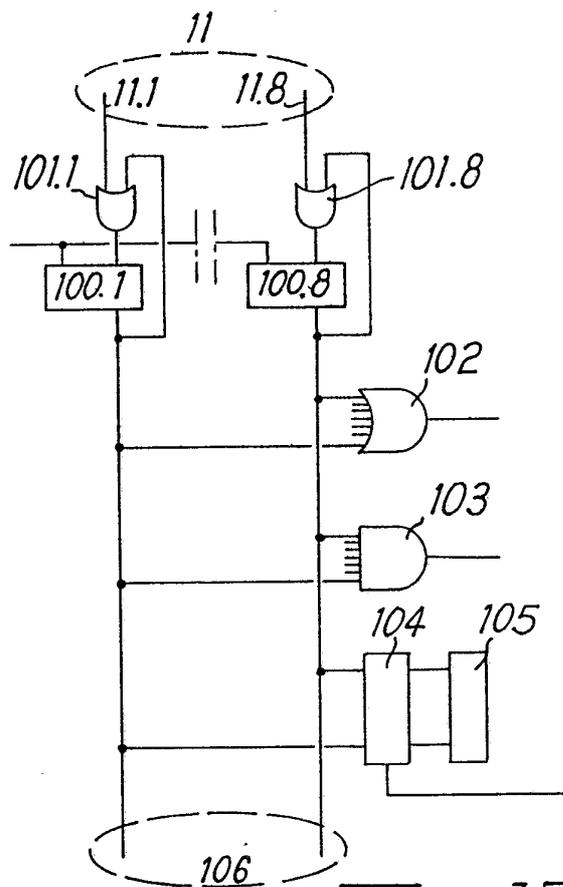


Fig. 16

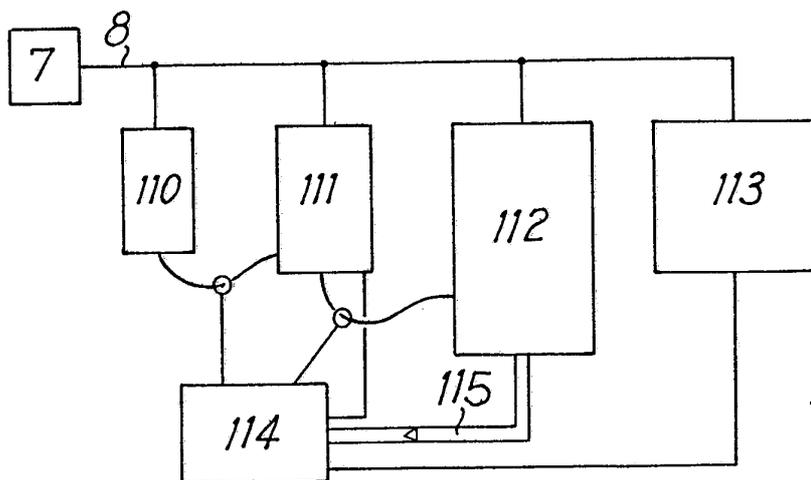
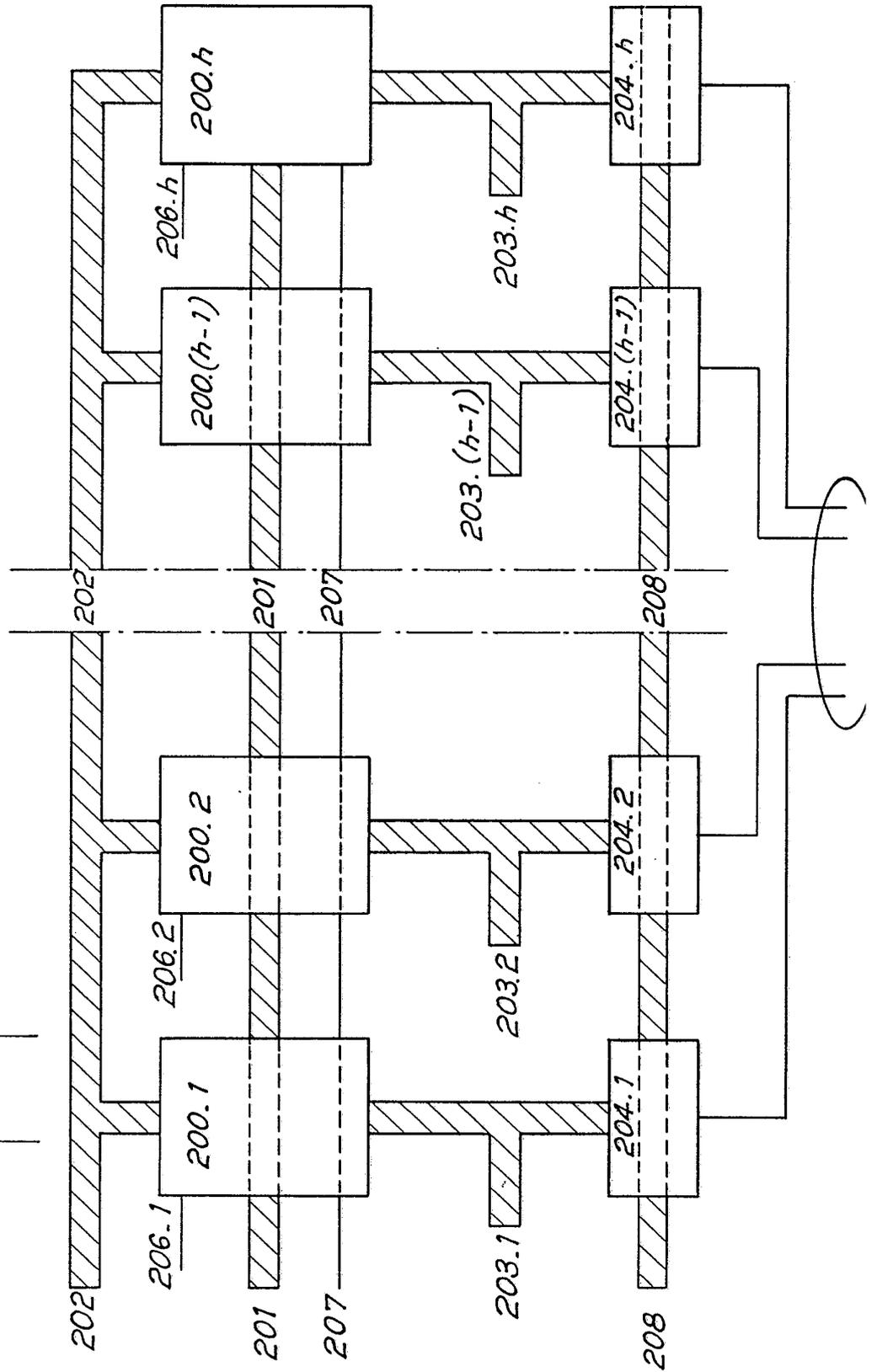


Fig. 17

FIG. 16



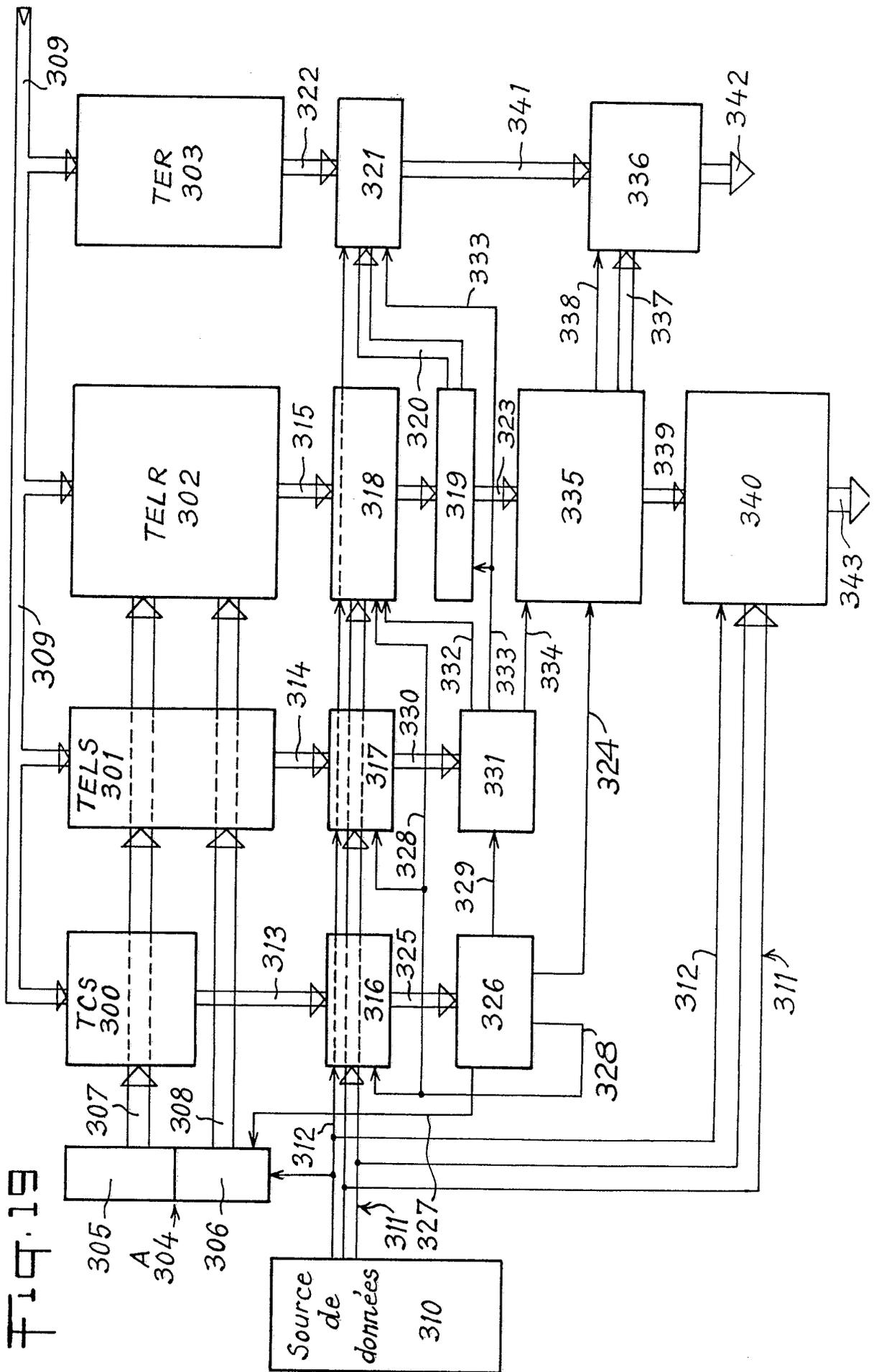


Fig. 20

