

A1

**DEMANDE  
DE BREVET D'INVENTION**

②①

**N° 81 12908**

---

⑤④ Dispositif logique programmable de prise, traitement et transmission de données.

⑤① Classification internationale (Int. Cl. <sup>3</sup>). G 05 B 19/18; H 03 K 19/173.

②② Date de dépôt..... 24 juin 1981.

③③ ③② ③① Priorité revendiquée :

④① Date de la mise à la disposition du  
public de la demande..... B.O.P.I. — « Listes » n° 52 du 31-12-1982.

---

⑦① Déposant : FORDAHL, société à responsabilité limitée, résidant en Suisse.

⑦② Invention de : Jean Engdahl et Jérôme Fomraz.

⑦③ Titulaire : *Idem* ⑦①

⑦④ Mandataire : Cabinet François Hagry, conseil en brevets d'invention,  
2, rue du 8-Mai, 74100 Annemasse.

---

La présente invention concerne les dispositifs électroniques ayant pour but de présenter des résultats ou provoquer des actions sur la base d'un certain nombre d'informations telles que grandeurs d'entrées, mode de traitement voulu, etc.

5 Pour atteindre ce but on a recours selon les cas :

- à des systèmes analogiques où les grandeurs peuvent prendre n'importe quelle valeur dans une gamme donnée (par exemple voltmètre),
- à une logique câblée où des conditions présentes ou absentes, posées à des entrées, produisent une ou plusieurs informations de sortie, les
- 10 traitements allant d'une simple opération telle que ET, OU, à des opérations arithmétiques (numériques) ou faisant appel à l'élément temps (systèmes séquentiels),
- à des systèmes comportant un logiciel (software) plus ou moins programmable ou modifiable et le plus souvent construits autour d'un micro-
- 15 processeur,
- à un ordinateur.

Les deux premières solutions sont spécialisées en ce sens qu'elles doivent être réalisées une fois pour toutes en fonction d'une application précise.

20 Les deux dernières peuvent être à usage multiple, un ensemble donné pouvant se prêter à plusieurs applications par modification de la programmation.

Il est évident qu'un dispositif pouvant se prêter à une multiplicité d'emplois est très intéressant puisqu'il peut être produit en série, même si les applications ne portent chacune que sur un nombre limité de

25 pièces.

Malheureusement, la programmation présente deux inconvénients majeurs :

- les systèmes faisant appel au microprocesseur demandent, pour chaque
- 30 nouvelle application, un temps parfois considérable de programmation et ce travail doit être fait par un personnel hautement qualifié;
- les ordinateurs faisant usage de langages plus évolués sont plus rapides à programmer et ne demandent pas forcément la connaissance de leur structure interne. Par contre, leur coût est notablement plus élevé et
- 35 ils ne se prêtent par conséquent qu'à des cas justifiant un tel investissement.

Le but de la présente invention est de réaliser un dispositif logique programmable permettant de prendre en compte un certain nombre de données, de les traiter et de délivrer des instructions ou grandeurs de sortie tout en évitant les inconvénients précités, le dispositif pouvant  
5 être programmé très rapidement, aisément modifié pour une application nouvelle, et ceci par des instructions simples, faciles à assimiler par des gens peu au fait du jargon informatique.

Un autre objectif de l'invention est de réaliser un dispositif tel que le nombre d'entrées et de sorties peut être augmenté à volonté, réa-  
10 gissant plus vite que les ensembles basés sur un programme séquentiel tel qu'il est utilisé dans un microprocesseur ou un ordinateur, et pouvant gérer simultanément plusieurs traitements sans que ceux-ci n'interfèrent les uns avec les autres.

L'invention est exposée dans la description qui suit et pour l'in-  
15 telligence de laquelle on se référera au dessin dont la figure 1 unique est une illustration schématique d'un dispositif selon l'invention.

Le dispositif selon l'invention est représenté schématiquement sur la figure 1.

Trois mémoires ou groupes de mémoires 1, 2 et 3 contiennent respec-  
20 tivement des mots-instructions, des mots-grilles et des paires d'adresses. Ces trois mémoires contiennent chacune le même nombre de mots, chaque mot étant composé d'un certain nombre de bits, généralement 4, 8 ou 16 dans l'état actuel des circuits à disposition. Chaque mot est repéré par son numéro d'ordre que l'on nommera "adresse-mémoire" dans ce qui  
25 suit.

Un générateur d'adresses cycliques 4 fournit les adresses-mémoires de la première à la dernière, puis recommence périodiquement à cadence rapide. Les trois mémoires sont connectées en parallèle, c'est-à-dire que le numéro d'adresse-mémoire est commun pour les trois mémoires 1, 2,  
30 3.

Un circuit 5 permet de limiter le balayage à une zone donnée, par exemple de l'adresse-mémoire no. 142 à l'adresse-mémoire no. 1017 ou tout autre numéro, selon le besoin. Ceci permet de stocker plusieurs programmes dans un même jeu de mémoires. Le système comporte en outre  
35 trois bus 6, 12, 15, c'est-à-dire trois groupes de lignes reliant entre eux les divers éléments.

Le bus-données 6 peut être branché à la mémoire-instruction 1 par une porte 7 ainsi qu'aux circuits d'entrée 8, de sortie 9, au comparateur 10, à des registres 11 ou d'autres périphériques, conformément à

ce qui va être expliqué plus loin.

Le bus-grille 12 est relié directement à la mémoire-grille 2.

La mémoire-adresses 3 est reliée, d'une part, à un décodeur interne 13 permettant d'adresser certains éléments propres au fonctionnement du 5 dispositif, d'autre part par une porte 14 au bus-adresses 15.

Le comparateur 10 permet de savoir si, à un instant donné, le mot-instruction lu de la mémoire 1 correspond à celui reçu sur le bus-donnée 6, en provenance d'un élément d'entrée 8, d'un registre 11 ou d'un autre périphérique.

10 Enfin, un circuit spécial 16 qui sera ci-après nommé drapeau d'égalité peut être levé ou abaissé en fonction du résultat fourni par le comparateur 10.

Pour mieux illustrer le fonctionnement de l'ensemble, nous prendrons un exemple non-limitatif dans lequel chaque mémoire peut contenir 2048 15 mots de 8 bits, et par conséquent où chaque bus 6, 12, 15 comporte 8 lignes. Dans cet exemple, nous considérerons que le comparateur 10 peut être commandé de manière à ce qu'il lève le drapeau d'égalité 16, donc donne une comparaison comme vraie, lorsqu'une nouvelle condition est remplie (mode OU) ou abaisse le drapeau d'égalité 16 si une comparaison 20 aboutit à l'inégalité (condition ET).

On peut montrer que, dans cette configuration, l'ensemble comparateur 10, drapeau d'égalité 16 donne une réponse positive (vraie) lorsqu'une somme de produits logiques est satisfaite. On pourrait de même faire un produit de sommes, mais on s'en tiendra au premier cas dans l'exemple 25 en question.

Lorsque le drapeau d'égalité 16 est levé, il ouvre la porte 14 du bus-adresse 15 qui va recevoir, au pas suivant de l'adresse-mémoire, une paire d'adresses 15A et 15B provenant de la mémoire 3. L'une de ces adresses 15A est celle d'une source d'information : groupe d'entrées 8, 30 registre 11, mémoire-instruction 1 ou autre, dont le contenu sera présenté sur le bus-données 6, et l'autre, 15B est la destination : groupe de sortie 9, registre 11, comparateur 10, ou autre.

La mémoire-grille 2 alimente le bus-grille 12 dont le rôle principal est de ne pas prendre en considération certains bits d'un mot qui, 35 dans l'exemple pris ici, en comporte huit. Ceci peut être utile aussi bien pour un groupe d'entrées dont seules quelques unes sont significatives pour une décision ou pour un groupe de sorties dont un seul ou quelques bits doivent être modifiés. Dans le cas des entrées, la grille intervient au niveau du comparateur 10 puisque c'est celui-ci qui détermine

si une condition est remplie ou non.

Ce fonctionnement peut être illustré par l'exemple suivant :

Mémoire : Bit no.	1	2	3	4	5	6	7	8
- instruction	1	1	0	0	0	1	0	0
5 - grille	1	1	1	0	0	1	0	0
Entrées (bus-donnée)	1	1	0	1	0	1	1	0

donc accepté :                    1 1 0 X X 1 X X    X étant 0 ou 1.

La grille valant "0" pour les bits 4-5-7 et 8, ceci signifie que la condition est admise comme réalisée si les bits d'entrée pour lesquels la grille vaut "1" correspondent à l'instruction contenue dans la mémoire 1.

On rencontrerait un tel cas dans une commande de machine si le fait d'appuyer sur un bouton Stop devait être pris en compte quels que soient les états des autres informations d'entrée. Il y aurait alors une instruction 10000000 et une grille 10000000 si le premier bit signifiait que le bouton Stop a été actionné et l'instruction suivante mettrait inconditionnellement toutes les sorties à zéro pour interrompre le processus en cours.

Les entrées 8 peuvent être soit un groupe d'éléments de commande, par exemple des boutons à disposition de l'opérateur, soit des éléments propres à la machine ou au processus : contacts de fin de course, capteurs divers, indicateurs de grandeurs physiques (vitesse, température, tension, etc.) soit des consignes données sous forme numérique (codeurs, clavier, périphérique à sortie digitale, etc.).

Dans l'exemple précité, une entrée numérique pourra comporter deux chiffres décimaux, puisque 8 bits permettent de les transmettre sous forme binaire codée décimale (BCD).

Les sorties 9 peuvent être des groupes de huit sorties indépendantes, par exemple chacune commandant l'enclenchement ou le déclenchement d'un élément tel que moteur, valve, témoin lumineux, etc., ou être également une information numérique. Dans ce cas, le nombre de chiffres à transmettre n'est pas forcément limité à deux dans le cas d'un bus à 8 bits, puisqu'il suffit de faire en sorte que l'élément de sortie soit muni d'un registre à décalage enregistrant deux chiffres de plus lorsqu'il est adressé plusieurs fois consécutivement, ce qui permet de diriger des informations numériques comportant un nombre quelconque de chiffres vers des périphériques tels qu'affichage, imprimante, registres, etc.

La même chose vaut pour les entrées qui peuvent fournir des infor-

mations numériques comportant un nombre quelconque de chiffres, par groupe de deux si le bus est à 8 bits.

Il en va encore de même pour ce qui a été nommé jusqu'ici registres 11. Ils peuvent enregistrer des informations unitaires sur 1 à 8 bits 5 grâce à la grille 2 ou les données numériques.

Il sera souvent utile de prévoir un ou plusieurs registres 11 adressables bit par bit, grâce à la grille 2, pour départager des situations identiques des entrées. Il se peut, en effet, que pour un même état des entrées, des décisions différentes puissent ou doivent être prises en 10 fonction de situations rencontrées précédemment.

Une séquence d'instructions peut par exemple se présenter comme suit:

Adresse- mémoire	Instruction	Grille	Adresse-	
			entrée	sortie
110	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1	5	1
15 111	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	6	2
112	0 1 0 0 0 0 0 0	0 1 1 0 0 0 0 0	7	2
113	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	5	1
114	0 0 0 0 0 0 0 1	1 1 0 0 0 0 0 1	0	3

Ce qui pourrait signifier dans une commande de machine :

20 110	SI tous les éléments du bloc d'entrée no. 5 = 0
111	ET que le compteur 6 est à 255
112	ET que le poussoir no. 2 du bloc d'entrée no. 7 est à 1, le poussoir no. 3 étant à 0
113	OU que le poussoir no. 1 du bloc d'entrée no. 5 = 1
25	quelle que soit la position des autres
114	ALORS mettre les sorties 1 et 2 du bloc 3 à 0 ET la sortie 8 à 1.

La sortie 114 n'est donc donnée que si l'équation logique

$$(110 \text{ ET } 111 \text{ ET } 112) \text{ OU } (113) = 1$$

30 est satisfaite, ce qui est indiqué par le drapeau d'égalité 16.

On voit que l'adresse-sortie no. 1 demande le comparateur 10 en opération OU, la no. 2 en opération ET, tandis que l'adresse-entrée no. 0 demande l'ouverture de la porte 7 qui mettra sur le bus-donnée 6 le contenu de la mémoire-instruction 1. Ceci actionnera le bloc de sortie no. 35 3 qui tiendra compte, en outre de l'état de la grille 2.

On notera que l'instruction 1 et la grille 2 se combinent en ET, bit par bit, le résultat permettant d'enclencher ou déclencher une sortie. Le résultat sera en général mémorisé.

Ce système présente donc un jeu d'instructions très complet avec

entre autres :

- contrôle de l'état de tout ou partie des bits d'un bloc d'entrée 8
- contrôle de l'état d'un registre 11
- combinaison de ces contrôles sous forme d'une somme de produits
- 5 - transfert d'une instruction en mémoire 1 vers une sortie 9 ou un registre 11
- transfert d'une entrée vers une sortie 9 ou un registre 11
- commande du changement de l'état de tout ou partie des bits d'un bloc de sortie 9.

10 On constate qu'un registre 11 peut fonctionner comme entrée et comme sortie.

Si des opérations arithmétiques sont désirées ou si un grand nombre de registres 11 est nécessaire ou encore si l'élément temps doit être pris en considération pour fixer la durée d'une action, temporiser, etc.,  
15 il peut être avantageux de décentraliser une partie de la programmation à un registre "intelligent".

Un exemple non-limitatif serait celui d'un bloc entrée-sortie 17 comportant un microprocesseur avec sa mémoire-programme morte (ROM) et une certaine capacité de mémoire temporaire vive (RAM) dans laquelle on  
20 crée le nombre de registres voulus. On peut alors prévoir un jeu d'instructions telles que :

- contrôler si un registre 11 est à 0 ou non
- mettre un registre 11 à zéro
- transférer un nombre dans un registre 11 depuis une entrée ou la mé-  
25 moire-instruction 1
- effectuer des transferts de registre à registre 11
- compter : incrémenter ou décrémenter
- compter ou décompter une fréquence pour fixer des durées, temporiser, mesurer la durée d'un évènement
- 30 - effectuer des comparaisons entre registres 11 et fournir le résultat :  
A = B, A > B, A < B
- effectuer des opérations arithmétiques.

Pour gagner du temps, on peut mettre les nombres à traiter sur le bus-donnée 6 et les instructions sur le bus-grille 12.

35 L'avantage de la décentralisation de ces opérations est qu'elles peuvent s'effectuer sans empêcher le déroulement de l'analyse cyclique 4 des adresses-mémoires 3, ce qui résulte en un notable gain de vitesse de traitement.

Par opposition, dans les dispositifs usuels à microprocesseur ou mi-

ni-ordinateur, un changement d'état des entrées crée une interruption, c'est-à-dire une demande d'accès à l'unité centrale qui, si elle l'accepte, est alors mobilisée exclusivement pour cette tâche.

Au contraire, le système selon l'invention est toujours disponible pour une ou plusieurs prises de décisions au cours de chaque cycle des adresses-mémoires. Si le programme contient par exemple 1000 instructions et que la fréquence d'analyse est de 1 MHz, cela signifie que chaque instruction est testée 1000 fois par seconde, l'action de sortie correspondante étant produite dans la microseconde qui suit une comparaison raison satisfaisante.

En ce qui concerne la programmation de l'ensemble, les mémoires 1, 2 et 3 peuvent être du type mémoire morte (ROM), mais on utilisera de préférence des mémoires mortes programmables par l'utilisateur (PROM) ou mieux, des mémoires mortes programmables et effaçables que l'on peut effacer et reprogrammer (EPROM).

Il est possible d'écrire les programmes sous la forme décrite dans l'exemple précédemment donné, mais on fera de préférence appel à un logiciel de développement qui permet très simplement d'établir le programme, voire de le transférer dans les mémoires 1, 2, 3 à partir d'instructions claires pour le non-initié.

On créera d'abord un vocabulaire mnémonique selon lequel on baptisera les entrées 8, les registres 11 et les sorties 9. On introduit alors les instructions sous forme d'équations logiques telles que :

$$\begin{aligned} & \text{AUT} = 0 * \text{FC} = 1 * \text{CTOP} = 0 \\ & + \text{STP} = 1 \\ & = \text{MOT} = 0 * \text{LT3} = 1 \end{aligned}$$

qui pourraient signifier : si le bouton AUT (automatique) n'est pas actionné ET que le capteur de fin de course FC1 est actionné ET que le compteur d'opérations CTOP est à zéro OU que le bouton STP (stop) est pressé ALORS couper le moteur ET allumer la lampe-témoin no. 3.

Le logiciel attribue les adresses d'entrée et de sortie 3 et crée les mots instruction 1 et grille 2 correspondants. Il peut également préparer directement les données nécessaires à la transcription dans des mémoires mortes programmables éventuellement effaçables (PROM, EPROM).

Le dispositif selon l'invention présente, outre ce qui a été dit, l'avantage que, n'étant pas séquentiel, les groupes d'instructions peuvent être introduits dans n'importe quel ordre. Il suffit qu'une condition ou un groupe de conditions soit suivi de l'action correspondante. Mais, chacun de ces groupes conditions-actions peuvent être logés n'im-



- 8 -

porte où en mémoire. Ceci est particulièrement utile lors du développement puisque l'on peut à tout moment ajouter de nouveaux éléments au programme ou en éliminer sans rien changer au reste.

Enfin, puisque l'analyse cyclique 4 peut être limitée 5 à une partie seulement des mémoires 1, 2, 3, ceci pouvant être modifié à volonté, il est possible de stocker plusieurs programmes avec, par exemple, divers modes de fonctionnement, des programmes d'auto-contrôle et de diagnostic en cas de panne, etc.

Au cas où un tel dispositif reçoit une nouvelle affectation, il suffit de reprogrammer les trois mémoires 1, 2, 3 sans aucune modification de composants ou de circuits.

Dans la description qui précède, ainsi que dans les revendications est utilisé le terme "grille". Ce terme est à peu près équivalent à celui de "masque" habituel dans le domaine des microprocesseurs. Un masque sert à ne prendre en compte que certains bits dans un mot alors que dans le cas présent la grille peut également être appliquée à des sorties mémorisées, ce qui implique des opérations plus complexes que la simple opération ET du masquage. En outre, le bus-grille 12 est également susceptible de véhiculer des instructions à destination de périphériques intelligents, ce qui est encore plus qu'un simple masquage. Aussi a-t-on choisi le terme "grille" de préférence à "masque".

REVENDEICATIONS

1. Dispositif logique programmable caractérisé par le fait qu'il comporte trois mémoires ou groupes de mémoires (1, 2, 3), la première contenant des mots-instructions (1), la seconde des mots-grille (2) permettant de prendre en considération tout ou partie d'un mot, la troisième des paires d'adresses (3), définissant des éléments-sources ou destinations à mettre en communication, et comportant en outre au moins un comparateur logique (10) permettant de savoir si une condition est réalisée en vue de la faire suivre d'une action, les mémoires (1, 2, 3) étant explorées cycliquement (4) et en synchronisme entre elles.
- 10 2. Dispositif selon la revendication 1, caractérisé par le fait qu'il comporte en outre un premier bus (6) ou groupe de lignes permettant de mettre en communication soit une entrée (8) avec le comparateur (10), un registre (11) ou un élément de sortie (9), soit la mémoire-instruction (1) avec un registre (11) ou un élément de sortie (9), et un deuxième bus (12) appliquant le mot-grille (2) au comparateur (10), à un registre (11) ou à un élément de sortie (9), et un troisième bus (15) amenant les adresses provenant de la mémoire-adresses (3) aux circuits d'entrée et de sortie (8, 9).
- 20 3. Dispositif selon l'une quelconque des revendications 1 et 2, caractérisé par le fait que le comparateur (10) arme un drapeau d'égalité (16) si une succession de comparaisons satisfait à une équation logique contenant des opérations ET et des opérations OU présentées sous forme d'une somme de produits ou d'un produit de sommes, ce drapeau (16) libérant alors l'instruction à donner (7) lorsqu'une certaine situation des entrées a ainsi été reconnue comme vraie.
- 30 4. Dispositif selon l'une quelconque des revendications 1 à 3, caractérisé par le fait qu'un ou plusieurs périphériques à programmes décentralisés (17) prennent en charge des opérations d'arithmétique, de comparaison, de comptage d'unités de temps ou toute autre impliquant un fonctionnement non-directement fourni par le système central, et ceci de manière à ne pas interférer avec le programme principal ni en retarder l'exécution.
5. Dispositif selon la revendication 1, caractérisé par le fait qu'un générateur d'adresses cycliques (4) opère l'analyse cyclique des 35 mémoires (1, 2, 3), cette analyse étant limitée entre une adresse initiale et une adresse finale par un limiteur de balayage (5), libérant ainsi d'autres parties de mémoire pour y loger des programmes secondaires ou d'aide au diagnostic en cas de défaut.

6. Dispositif selon la revendication 1, caractérisé par le fait qu'un logiciel approprié (17) permet de présenter les conditions et décisions sous forme mnémonique et les traduit directement en instructions de programmation pour les mémoires.

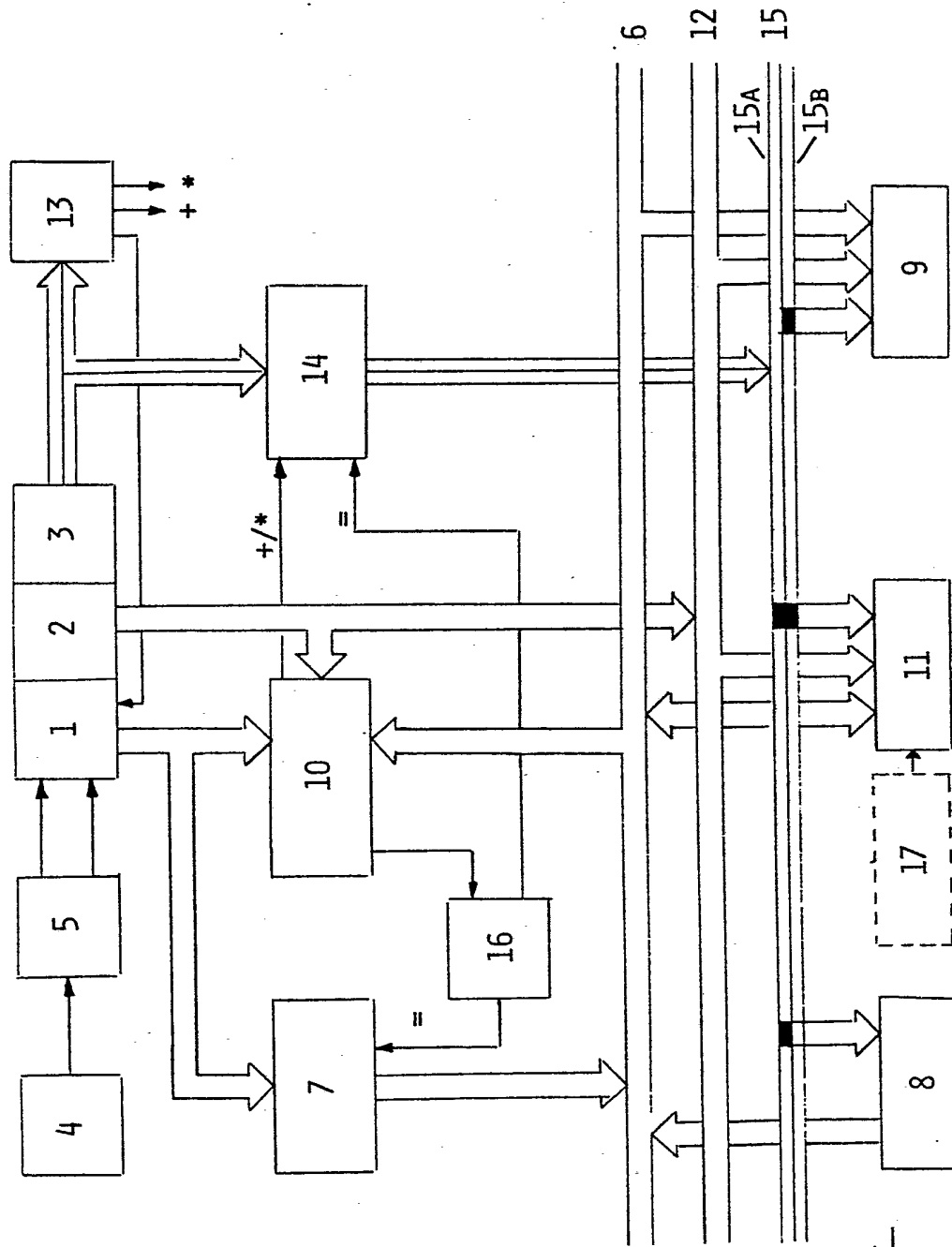


FIG. 1