



CONFÉDÉRATION SUISSE
OFFICE FÉDÉRAL DE LA PROPRIÉTÉ INTELLECTUELLE

Int. Cl.³: G 05 B 19/18
G 06 F 15/46

Brevet d'invention délivré pour la Suisse et le Liechtenstein
Traité sur les brevets, du 22 décembre 1978, entre la Suisse et le Liechtenstein



FASCICULE DU BREVET A5

(11)

637 228

(21) Numéro de la demande: 2441/80

(73) Titulaire(s):
Willemin Machines S.A., Bassecourt

(22) Date de dépôt: 27.03.1980

(72) Inventeur(s):
Jean-Pierre Voillat, Delémont

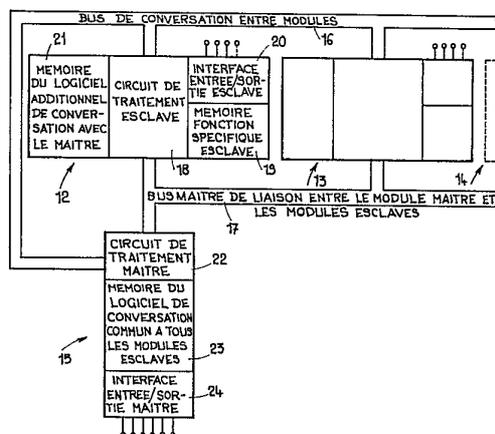
(24) Brevet délivré le: 15.07.1983

(45) Fascicule du brevet
publié le: 15.07.1983

(74) Mandataire:
Jean S. Robert, Landecy-Genève

(54) Dispositif de commande d'une machine ou d'une installation.

(57) Le dispositif comprend plusieurs modules esclaves (12, 13, 14) destinés chacun à gérer une fonction particulière et un module maître (15) destiné à gérer l'ensemble des modules esclaves. Chacun des modules esclaves comprend un circuit de traitement (18) coopérant avec un interface entrée/sortie (20) pour permettre la conversation avec l'opérateur et un circuit de fonction (19) destiné à spécifier la fonction que doit gérer le module esclave, tandis que le module maître (15) comprend un circuit de traitement (22) coopérant avec une mémoire (23) pour le logiciel de conversation et de traitement des informations reçues par les modules esclaves, et avec un interface entrée/sortie (24) pour assurer la conversation avec l'opérateur. L'ensemble des modules sont interconnectés d'une part par un bus (16) de conversation entre les modules et d'autre part par un bus maître (17) destiné à permettre au module maître de venir lire directement dans une mémoire (21), que comprend chacun des modules esclaves, le logiciel additionnel nécessaire à la conversation avec le module esclave.



REVENDEICATIONS

1. Dispositif de commande d'une machine ou d'une installation, comprenant un module maître (15) et plusieurs modules esclaves (12, 13, 14), le module maître étant destiné à assurer la gestion de l'ensemble du dispositif et les modules esclaves à assurer la gestion de fonctions particulières, caractérisé par le fait que chaque module esclave (12, 13, 14) comprend des moyens (21) lui permettant d'emmagasiner la portion de logiciel dont le module maître a besoin pour converser avec lui et pour traiter les informations reçues.

2. Dispositif suivant la revendication 1, caractérisé par le fait qu'il comprend un bus (16) de conversation directe, bidirectionnelle, entre n'importe quels modules (12, 13, 14, 15) et un bus maître (17) de liaison entre le module maître (15) et les modules esclaves (12, 13, 14) pour permettre au module maître de chercher directement dans chaque module esclave la portion de logiciel contenue dans lesdits moyens (21) pour converser avec lui.

3. Dispositif suivant la revendication 2, caractérisé par le fait que tous les modules esclaves (12, 13, 14) comprennent un circuit de traitement (18), le même pour tous, une mémoire de fonction (19), spécifique à chaque module esclave, et un interface entrée/sortie (20) également spécifique à chaque module esclave, le module maître comprenant un circuit de traitement maître (22), une mémoire (23) du logiciel de conversation commun à tous les modules esclaves et un interface entrée/sortie (24).

4. Dispositif suivant la revendication 3, caractérisé par le fait que chacun des modules est monté séparément dans un tiroir comprenant un panneau portant les interfaces entrée/sortie de conversation avec l'opérateur, le panneau du tiroir du module maître (80) portant un clavier permettant d'entrer dans le dispositif, lors de l'introduction du programme, les données relatives aux séquences de ce programme, un clavier de dix touches permettant d'entrer les valeurs numériques de toutes les opérations effectuées par les modules esclaves, ainsi qu'un premier (OPER) et un deuxième (SEQUENCE) affichage permettant d'afficher, lors du déroulement du programme, respectivement l'opération maître et le numéro de la séquence en cours, et par le fait qu'au moins une partie des tiroirs esclaves (81, 82, 83, 84) comprennent des touches permettant d'entrer dans le programme les opérations qui doivent être exécutées par le tiroir esclave dans la séquence en cours.

5. Dispositif suivant la revendication 3, caractérisé par le fait qu'il comprend un clavier unique de conversation avec les modules.

6. Dispositif suivant la revendication 1, caractérisé par le fait que chacun des modules comprend un circuit de traitement à microprocesseur (25), au moins un circuit de fonction spécifique (29), une logique d'interruption (26), une mémoire morte (27) et une mémoire vive (28) interconnectés par un bus d'adresses interne (30a), un bus de données interne (30b), un bus de commande interne (30c) et un bus d'interruption interne (30d), une mémoire morte additionnelle (31) qui, dans le cas du module maître seulement, est reliée par un bus additionnel (38) au bus de données interne (30b) et qui, pour tous les modules, est reliée à un bus d'adresses extérieur maître (34) et à un bus de données extérieur maître (35) pour permettre au module maître de lire ladite portion de logiciel contenue dans ladite mémoire morte additionnelle (31).

L'invention a pour objet un dispositif de commande d'une machine ou d'une installation, comprenant un module maître et plusieurs modules esclaves, le module maître étant destiné à assurer la gestion de l'ensemble du dispositif et les modules esclaves à assurer la gestion de fonctions particulières. On connaît déjà de tels dispositifs à traitements multiples (multi processing) dans lesquels chaque module esclave possède le logiciel (software) relatif à la fonction qu'il doit gérer et le module maître possède le logiciel lui permettant

de converser avec tous les modules esclaves de façon à gérer l'ensemble du dispositif de commande. Dans ces dispositifs connus, il est impossible d'ajouter un module esclave d'un nouveau type sans devoir reconsidérer tout ce logiciel. De plus, ces dispositifs à traitements multiples connus sont lents car, chaque fois qu'un module veut appeler, il doit tout d'abord remplir une mémoire générale, avertir le module appelé qu'il y a un message pour lui, ce dernier vidant la mémoire traitant les informations reçues, puis donnant sa réponse en remplissant à nouveau la mémoire générale, en avertissant le module appelant, etc.

Le but de l'invention est d'éliminer les inconvénients ci-dessus constitués, d'une part, par la difficulté à ajouter un nouveau module esclave au dispositif, en obligeant chaque fois l'opérateur à revoir le logiciel maître et, d'autre part, par la lenteur relative du dispositif.

Selon l'invention, le dispositif est caractérisé par le fait que chaque module esclave comprend des moyens lui permettant d'emmagasiner la portion de logiciel dont le module maître a besoin pour converser avec lui et pour traiter les informations reçues.

En d'autres termes, chaque module esclave met lui-même à disposition du module maître la portion de logiciel permettant la gestion de ce module esclave par le module maître.

Le dessin illustre un dispositif à traitements multiples connu et, à titre d'exemple, une forme d'exécution du dispositif selon l'invention:

la fig. 1 est le schéma bloc du dispositif à traitements multiples connu,

la fig. 2 est le schéma bloc correspondant d'un dispositif de commande à traitements multiples selon l'invention,

les fig. 3A et 3B représentent ensemble le schéma bloc d'un module du dispositif selon l'invention,

la fig. 4 est un diagramme représentant différents signaux apparaissant dans les circuits, et

la fig. 5 illustre un exemple d'utilisation du dispositif, pour la commande d'une machine-outil.

Le schéma bloc de la fig. 1 représente un dispositif de commande connu d'une machine-outil ou d'une installation qui comprend plusieurs modules esclaves 1, 2, 3, 4... Chacun de ces modules esclaves est destiné à commander un organe particulier de la machine-outil, par exemple, dans une fraiseuse, un moteur d'avance de la table selon l'axe des X, un moteur d'avance de la table selon l'axe des Y, un moteur d'avance de la broche, etc. Chacun de ces modules 1-4 comprend un circuit de traitement esclave 5, une mémoire esclave 6 destinée à emmagasiner les données propres à la commande de l'organe correspondant et un interface entrée/sortie assurant la liaison entre le module et l'organe correspondant. Pour assurer la synchronisation de tous ces modules esclaves, le dispositif comprend encore un circuit de traitement maître 8, une mémoire générale 9 et un interface général entrée/sortie 10, le tout interconnecté par un bus général 11. Chaque fois que l'un des modules esclaves veut converser avec un autre module esclave, il doit d'abord charger la mémoire générale et avertir le module appelé qu'un message l'attend. Le module appelé décharge la mémoire, analyse les informations, puis donne sa réponse en rechargeant la mémoire générale, en avertissant le premier appelant, etc. Il est clair qu'un tel processus est lent. Mais un inconvénient plus grave encore de ce système réside dans le fait que le circuit maître 8, la mémoire générale 9 et l'interface général 10 doivent prévoir au départ toutes les possibilités offertes par un nombre variable de modules esclaves, ce qui nécessite un langage d'informatique qui n'est pas à la portée d'un opérateur de machine-outil et, surtout, empêche le constructeur d'augmenter ou de diminuer la capacité du système sans modification importante de l'appareillage.

La fig. 2 est un schéma bloc illustrant la présente invention. Le dispositif représenté comprend des modules esclaves 12, 13, 14 et un module maître 15, tous les modules étant reliés entre eux par un bus de conversation 16 et par un bus maître 17 de liaison entre le module maître et les modules esclaves. Chaque module esclave comprend un circuit de traitement esclave 18, une mémoire 19 de fonction spécifi-

que esclave et un interface entrée/sortie esclave 20, ces trois parties correspondant en gros aux parties 5, 6, 7 de la fig. 1. Mais ils comprennent en plus une mémoire 21 du logiciel additionnel de conversation avec le maître qui, comme son nom l'indique, est destinée à renfermer la partie du logiciel de conversation propre au module esclave correspondant. Le module maître 15 comprend un circuit de traitement maître 22, une mémoire 23 du logiciel de conversation avec les modules qui est commun à tous ces modules de même qu'un interface entrée/sortie maître 24. Ainsi donc, la partie du logiciel de conversation commune à tous les modules esclaves est contenue dans la mémoire maître, tandis que la partie de ce logiciel qui est propre à un module esclave particulier est contenue dans ce module esclave lui-même. Ainsi, si l'on veut passer d'un dispositif simple comprenant peu de modules esclaves à un dispositif en comprenant davantage, il suffit d'ajouter chaque fois le logiciel de conversation additionnel dans les modules esclaves supplémentaires, où le module maître ira le chercher.

Les fig. 3A et 3B représentent ensemble le schéma bloc d'un module, les parties en pointillé ne concernant que le module maître et devant être supprimées pour les modules esclaves.

Les modules comprennent un circuit de traitement à micro-processeur 25, un circuit logique d'interruption 26, une mémoire morte (ROM) 27, une mémoire vive (RAM) 28 et un circuit de fonction spécifique 29, ces éléments étant reliés par un bus d'adresses interne 30a, un bus de données interne 30b, un bus de commande interne 30c et un bus d'interruption interne 30d. Tous les bus 30a à 30d mentionnés ci-dessus sont internes au module et ne figurent par conséquent pas sur la fig. 2. Ce circuit de fonction spécifique fait du module considéré un organe de commande d'un dispositif déterminé. Ce circuit n'est que l'interface entre le microprocesseur dudit module et l'organe final du dispositif déterminé.

La mémoire du logiciel additionnel de conversation avec le module maître, qui est représentée en 21 à la fig. 2, est constituée par une mémoire morte (ROM) 31 (fig. 3B) où l'on voit qu'elle coopère avec un miniswitch de codage 32 et avec un premier comparateur 33a, un second comparateur 33b étant spécifique à la conversation intermodules uniquement. Les éléments 31 et 33a des différents modules sont reliés l'un à l'autre par un bus d'adresses maître 34, un bus de données maître 35, une ligne de commande de mémoire 36, ces trois éléments étant représentés globalement à la fig. 2 par le bus 17. Dans le cas où le module représenté est maître, il est clair que ses bus internes 30a, 30b, 30c sont reliés aux bus externes correspondants 34, 35, 36 par un circuit d'actionnement (driver) 37, un bus 38 et une ligne de commande 39. On voit ainsi que le logiciel additionnel de conversation avec le maître pourra être transmis à ce dernier, de la mémoire morte 31 du module esclave au processeur maître uniquement, lors de ses demandes d'instructions. De même, le module maître pourra communiquer les adresses et un ordre de lecture de mémoire par les circuits 37, le bus d'adresses maître 34 et les lignes 36 et 39.

Pour pouvoir assurer la conversation directe entre modules, le circuit de chaque module comprend deux flip-flops RS 40, 41 deux décodeurs 42, 43, une porte bidirectionnelle 44, une porte unidirectionnelle 45, un circuit logique de test 46 et deux portes à trois états (tristate) 47, 48. Ces éléments sont interconnectés, d'une part, entre eux et, d'autre part, par un ensemble de six lignes de commande 49, 50, 51, 52, 53, 54, par un bus de données de conversation 55 et par un bus d'adresses de conversation 56, avec les autres modules. Les éléments 49 à 56 sont, comme on le voit à la fig. 3, externes aux modules et constituent en fait le bus de conversation 16 représenté à la fig. 2 pour la liaison directe des modules esclaves entre eux.

L'interconnexion des éléments internes au module est la suivante: le circuit 25 reçoit un signal READY des sorties \bar{Q} des flip-flops 40 et 44 par une ligne 57, et applique lui-même un signal STROBE au décodeur 42 par une ligne 58. Ce dernier applique un signal ACWR à l'entrée S du flip-flop 41, par une ligne 59, un signal ACRD à l'entrée S du flip-flop 40 par une ligne 60 et un signal ACT au circuit de test 46 par une ligne 61. Le décodeur 43 envoie, aux

entrées R des flip-flops 40 et 41 des autres modules, des signaux CWR et CRD par des lignes 62, 64, 49, respectivement 63, 65, 50. Le circuit de test 46 reçoit et/ou émet des signaux CBUSY, SLI et SLO par des lignes 66, 67, 68 respectivement, signaux qui sont également appliqués aux entrées correspondantes des circuits de tous les modules par les lignes de bus 51, 52, 53, la sixième ligne 54 du bus de commande recevant un signal CINT par une ligne 69 de la porte à trois états 48. Concernant les lignes de bus 52 et 53, il faut noter qu'il s'agit en fait d'une même ligne dans laquelle les entrées 67 et sorties 68 sont reliées en série, le signal d'entrée SLI reçu par la ligne 52 étant constitué par le signal SLO du module amont, et le signal SLO envoyé par la ligne 53 constituant le signal SLI du module aval. Les signaux CINT des portes 48 de tous les modules sont appliqués au comparateur 33b de tous les modules par une ligne 70. La porte à trois états 47 applique un signal de sortie CINT à un bus 71 reliant la porte 44 au bus extérieur 55, les deux portes 47 et 48 étant commandées par deux signaux d'entrée COST et RCT émis respectivement par le circuit de test 46 et par le décodeur 43, sur des lignes 72, respectivement 73. Ce signal COST est également appliqué à la porte 45 par une ligne 74 et au décodeur 43 par une ligne 75, ce dernier appliquant un signal CEND au circuit 46 par une ligne 76. Le comparateur 33b émet un signal ICL qui est entré dans le bus 30d par une ligne 77 et dans le bus 71 par une ligne 78. Le comparateur 33a reçoit, par une ligne 79, un signal MEMR et un signal correspondant par la ligne 39, lorsqu'il s'agit du module maître.

Le fonctionnement du dispositif de commande est le suivant: les lignes et bus internes aux modules assurent la communication avec le circuit de fonction spécifique 29 correspondant, le bus d'interruption 30d en permettant au circuit de fonction spécifique 29 d'intrompre le circuit 25 dans son travail pour une communication importante, le bus d'adresses interne 30a en permettant au circuit 25 d'écrire ou de lire dans un organe déterminé du circuit 29, le bus de données 30b en véhiculant les mots de la conversation entre le circuit 25 et le circuit 29, et le bus de commande interne 30c en permettant la gestion de la conversation entre le circuit 25 et le circuit 29.

Les bus et lignes externes permettent la communication entre les modules, les lignes 49 à 54 en permettant la gestion des appels et des conversations entre les modules, le bus d'adresses 56 en permettant la mise en fonction conversationnelle du module appelé et le bus de données 55 en véhiculant les mots de la conversation entre les modules, le bus d'adresses maître 34 en permettant la lecture par le module maître de son logiciel contenu dans les modules esclaves, le bus de données maître en véhiculant les mots du logiciel maître et la ligne 36 en ouvrant au moment voulu la mémoire 31 en lecture. Une ligne de mise à zéro, non représentée, permet, lors de l'enclenchement de la machine, l'initialisation de tous les organes.

Le logiciel additionnel de conversation avec le module maître, que renferme chacun des modules esclaves, est contenu dans la mémoire morte 31; c'est pour cette raison que, lorsque le module représenté à la fig. 3 est un module esclave, les liaisons en pointillé sont à supprimer, la mémoire 31 étant lue par le bus de données maître extérieur 35. La mémoire 31 répond à un appel lorsque l'adresse transmise sur le bus d'adresses maître 34 correspond à l'adresse du module appelé qui est inscrite dans le miniswitch de codage 32, la comparaison étant effectuée dans le comparateur 33a. Le code introduit dans le miniswitch permet donc de différencier chaque module, aussi bien pour une conversation maître-esclave que pour une conversation entre esclaves. Dans le cas du module maître, le bus 38 et la ligne 39 sont en fonction de même que le circuit d'actionnement (driver) 37; de ce fait, ce module sera maître du bus d'adresses et, par-là, de toutes les mémoires mortes 31.

La fig. 4 représente de façon détaillée le rôle de différents signaux dans une conversation entre modules esclaves, en A lorsque la ligne 51 pour le signal CBUSY (conversation en train) est libre au moment de l'appel, en B dans le cas où la ligne est libre mais où deux appels se produisent simultanément, et en C dans le cas où la ligne est occupée au moment de l'appel.

Une conversation est demandée par le circuit de traitement 25, dans un seul cycle machine, ce dernier envoyant au décodeur d'entrée/sortie 42 un signal de lecture d'entrée/sortie (non représenté) par l'intermédiaire du bus de commande 30c. Dès l'apparition — au tout début d'un cycle machine — de ce signal, le décodeur 42 émet un signal \overline{ACT} (Advance Conversation Test) sur la ligne 61, ce qui provoque la réponse instantanée du circuit logique de test 46.

Si la ligne 51 n'est pas occupée (cas de la fig. 4A), le signal CBUSY (Conversation Busy) se trouve à 1. Dans l'infime instant qui suit, et pour autant que la ligne SLI (Security Line In) soit aussi à 1, le signal CBUSY tombe à 0 sur ordre du circuit 46 pour marquer l'établissement d'une conversation, et le signal SLO (Security Line Out) tombe également à 0 sur ordre du circuit 46 pour assurer qu'aucun module en aval n'ait pu se connecter simultanément en conversation. Dès lors, le circuit 46 fait passer à 0 le signal \overline{COST} (Conversation Start) de la ligne 72, indiquant ainsi au circuit de traitement (processeur) 25 que la conversation est établie, ainsi que cela apparaîtra plus loin. Dans le cas où les lignes sont occupées au moment de l'appel (fig. 4C), le signal \overline{ACT} ne provoque, tant sur les lignes que sur le circuit 46, aucune modification et le signal \overline{COST} demeure à 1, indiquant au circuit 25 que la conversation n'est pas établie. Ce signal \overline{COST} demeure présent tout au long de la conversation et, dès son apparition, provoque l'application sur le bus d'adresses 56 de l'adresse du module appelé, en agissant sur la porte 45 par la ligne 74. Ainsi, la porte 45 est ouverte, ce qui permet d'avoir immédiatement une quittance du module appelé quant à sa possibilité d'entrer en conversation, ce dernier pouvant par exemple se trouver en interdiction d'interruption. Quelques fractions de μ s après l'apparition du signal \overline{ACT} , le décodeur 43 émet un signal \overline{RCT} (Read Conversation Test) sur la ligne 73. Pendant ce signal, le circuit de traitement 25 emmagasine les informations présentes sur son propre bus de données 30b. Les fonctions de l'impulsion \overline{RCT} sont multiples. Nous avons simultanément:

1. L'envoi par la porte 48 d'un signal \overline{CINT} (Conversation Interrupt) sur la ligne 69, signal dont la valeur est 0, provoque, sur le module répondant à l'adresse demandée par la porte 45, une réquisition d'interruption. Ainsi, à l'apparition d'un 0 sur la ligne 70 du module appelé et, si le comparateur 33b reconnaît une égalité entre l'adresse reçue du bus 56, d'une part, et du switch de codage 32, d'autre part, et pour autant que le circuit 25 du module appelé accepte l'interruption, le comparateur 33b génère un signal \overline{ICL} (Interrupt Conversation Line) de valeur 0, sur les lignes 77 et 78, mettant ainsi le circuit de traitement 25 du module appelé en interruption (Interrupt) et préparant la quittance sur le bus 55 à l'intention du circuit 25 du module appelant.

2. En ouvrant la porte 47, le signal \overline{RCT} introduit sur un bit du bus 55 la quittance selon laquelle la conversation est engagée.

3. Le signal \overline{RCT} , transmis directement du décodeur 43 à la porte bidirectionnelle par les bus les reliant, provoque l'ouverture de cette dernière de façon à brancher le bus 55 sur le bus 30b dans le module appelant.

4. Enfin, à la fin de l'impulsion du signal \overline{RCT} , le circuit 25 appelant emmagasine les informations transmises par la porte 44, ce qui permet à ce dernier d'interpréter immédiatement les deux bits de quittances, apprenant du même coup si la conversation est branchée et si le circuit de traitement 25 appelé est prêt à la conversation.

Si, par contre, lors du passage de l'impulsion du signal \overline{ACT} , la conversation n'a pu s'établir (cas des fig. 4B et 4C), le signal \overline{COST} reste à 1 et le décodeur 43 ne peut générer le signal \overline{RCT} . Il n'y aura dans ce cas aucune demande d'interruption et la porte 43 restera fermée. Le circuit 25 appelant lira tout de même la porte 44 qui ne lui fournira que des 1 et tirera la conclusion que la conversation n'est pas établie.

Rappelons que tout ce qui vient d'être décrit n'occupe qu'un cycle machine.

Admettons maintenant que la conversation a été établie. Cela signifie tout d'abord que chacun des deux modules est en train d'exécuter le programme régissant cette conversation: le module appe-

lant, parce qu'il a choisi lui-même de le faire, et le module appelé, parce qu'il a été forcé de le faire suite à la demande d'interruption reçue. Il est clair que ces deux programmes devront avoir été conçus de façon à réaliser la même conversation.

Pour illustrer ce qui va se passer, imaginons une conversation fort simple: le module appelant X envoie une information, dont il attend la réponse, au module appelé Y.

Considérons tout d'abord le module X; il doit envoyer son message et le fera par l'intermédiaire de l'impulsion du signal \overline{ACWR} sur la ligne 60 et de l'impulsion du signal \overline{CWR} sur la ligne 62. Le signal \overline{ACWR} va, dès son apparition et grâce au flip-flop 41, préparer le circuit 25 à une mise en attente qui deviendra effective dès l'apparition de l'impulsion du signal \overline{CWR} qui ouvre la porte bidirectionnelle dans le sens circuit 25, X vers l'extérieur. Le message prend donc la direction du circuit 25, Y.

Laissons le circuit 25, X en attente et voyons ce que fait le module Y. Celui-ci, de par son programme régissant cette conversation, sait qu'il doit s'ouvrir au message du module X. Il le fait par l'intermédiaire des impulsions de signaux \overline{ACRD} (Advanced Conversation Read) sur la ligne 60 et \overline{CRD} (Conversation Read) sur la ligne 65. Le signal \overline{ACRD} , dès son apparition et grâce au flip-flop 40, prépare le circuit 25 du module Y, à la mise en attente. L'attente commence avec le début de l'impulsion du signal \overline{CRD} qui ouvre la porte 44 dans le sens extérieur vers le circuit 25, Y.

Maintenant les deux circuits de traitement 25 sont en attente. L'impulsion du signal \overline{CWR} du module X, qui est présente pendant toute l'attente, est envoyée, à travers la ligne extérieure 49, sur le flip-flop 40 du module Y. Ce flip-flop, précisément, empêche le module Y de continuer à travailler et cette impulsion du signal \overline{CWR} émise par le module X va remettre à 0 la sortie Q du flip-flop 40 et permettre au module Y de continuer à travailler. De même, l'impulsion du signal \overline{CRD} émise par le module Y et présente pendant toute l'attente de celui-ci aboutit au flip-flop 41 du module X qui empêche ce module de continuer à travailler. L'impulsion du signal \overline{CRD} du module Y va permettre au module X de continuer à travailler.

Ainsi nous voyons que quel que soit le retard ou l'avance des deux circuits de traitement 25 l'un par rapport à l'autre, dès qu'il y a un mot à passer, ils vont s'attendre mutuellement. Lorsque les circuits 25 des deux modules sont en position d'émettre, respectivement de recevoir, ils vont tout naturellement passer le message.

Pour résumer, dès qu'un circuit 25 veut émettre ou recevoir un mot, il s'autobloque; il sera libéré par l'autre circuit 25 aussitôt que celui-ci est prêt à recevoir, respectivement à émettre.

Cela permet donc, par un seul cycle machine, de transmettre un mot, puisque chaque circuit 25 traite séparément le message passé et se remet en position d'échange d'information sans tenir compte du partenaire puisque, de toute façon, le message passera dès que chacun aura terminé le petit travail qui l'occupe.

La conversation sera terminée, dès que le logiciel du module appelant en décidera, par un simple cycle machine fournissant une impulsion sur un signal \overline{CEND} (Conversation End).

La fig. 5 illustre un exemple d'utilisation du dispositif dans la commande d'une machine à programme comprenant un chariot commandé par deux moteurs pour être amené en un point quelconque de coordonnées X, Y pour y effectuer différentes opérations.

Le dispositif comprend cinq modules logés dans des tiroirs dont on n'a représenté que les panneaux avant portant les interfaces de conversation avec l'opérateur et qui sont:

- un module maître 80 destiné à effectuer la gestion généralisée du programme,
- un module de fonctions 81 destiné à enclencher ou déclencher différentes fonctions telles que vannes, pompes, lampes, relais, etc.,
- deux modules 82 et 83 de commande de deux moteurs le long de l'axe X, respectivement Y, et
- un module cassette 84 permettant la lecture et le traitement des informations de la bande magnétique.

Les panneaux frontaux des tiroirs sont standardisés de manière à permettre l'aménagement de sept éléments d'affichage au maximum, au haut du panneau, et d'un clavier de vingt-quatre touches de commande au maximum, ces éléments correspondant à l'interface entrée/sortie 24 de la fig. 2. Les éléments d'affichage 85 sont constitués par des LED à sept segments et les touches de commande 86 portent un témoin 87 et une inscription 88 relative à leur fonction.

Le tiroir MAITRE 80 porte un clavier numérique composé de dix touches portant les inscriptions 0 à 9 et servant à introduire un nombre dans le dispositif, lors de la programmation, les autres tiroirs 81 à 84 ne comprenant pas de clavier numérique.

Dans la description ci-après, les touches sont désignées par l'inscription qu'elles portent. Elles servent, lors de la programmation :

SO	à la mise en position d'une séquence déterminée.
SQ+1	à l'incrémentement d'un compteur de séquences.
SQ-1	à la décrémentation de ce compteur.
Imp	à l'introduction d'un saut d'une séquence à une autre.
CALL	à l'introduction de sous-programmes répétitifs dans le programme d'usinage d'une pièce, par exemple pour effectuer un calcul.
RET	à quitter un sous-programme répétitif pour retourner dans le programme principal.
END	à marquer la fin du programme.
TEST	à contrôler le déroulement d'un programme de façon fictive, uniquement par simulation des affichages.
TEMP	à ajouter une temporisation, une fois la séquence terminée, avant de passer à la suivante.
CLR	à annuler le contenu total d'une séquence pour effectuer, par exemple, une correction.
CE	à n'annuler que le dernier ordre introduit.

Enfin, l'affichage se divise en deux groupes, l'un marqué OPER pour désigner le numéro de l'opération JUMP, CALL, RET, etc..., et l'autre, marqué SEQUENCE, composé de quatre LED, dans lequel est affiché le numéro de la séquence dans laquelle est effectuée l'opération désignée par le premier groupe.

Le tiroir FONCTIONS 81 comprend un groupe de seize touches portant les inscriptions F0 à F15 désignant le numéro de la fonction que la machine doit exécuter, un groupe de deux touches portant les inscriptions IN, respectivement OUT, un affichage à une LED marquée IN/OUT et un affichage à deux LED marquées FONCTION. Les touches permettent d'entrer le programme, comme on le verra plus loin, l'affichage FONCTION indique le numéro de la fonction en cours d'exécution et l'affichage IN/OUT si cette fonction est enclenchée ou déclenchée.

Les deux tiroirs MOTEUR 82, 83 servent à commander les deux moteurs de positionnement X, respectivement Y. Ils comprennent neuf touches permettant d'entrer le programme des moteurs et dont les fonctions sont les suivantes :

OR	sert à commander le retour à l'origine.
+N	sert à avancer d'un nombre donné.
-N	sert à reculer d'un nombre donné.
+CR	sert à effectuer une correction d'outil positive.
-CR	sert à effectuer une correction d'outil négative.
VT	sert à entrer la vitesse d'avance.
EXT ₀ , EXT ₁ , EXT ₂	servent à prendre en considération des coordonnées fournies par un autre module, par exemple le module cassette.

L'affichage unique à sept LED, marqué VALEUR, indique la valeur numérique de l'opération en cours et qui a été entrée, rappelons-le, par le module MAITRE, par exemple la valeur de la vitesse d'avance, la valeur de la correction d'outil, la valeur du nombre +N ou -N, etc.

Le tiroir CASSETTE 84 est un tiroir classique mais comprenant en plus l'électronique permettant la liaison de conversation entre tiroirs et la liaison maître avec le tiroir maître.

Le panneau comprend un mécanisme d'insertion des cassettes 89 et quatre touches.

La cassette est enregistrée par portions qui toutes présentent, à la fin, un signal d'arrêt. Ces portions sont appelées selon la programmation, dans les séquences voulues, par le module maître.

Le programme de commande de la cassette est entré par les trois touches du bas qui servent :

START à commander la lecture, à une séquence donnée, de la première portion.

à commander la lecture de la portion suivante.

REW à commander le rebobinage de la cassette.

La touche marquée EJECT ne sert pas à entrer le programme et ne porte pas de témoin; elle sert simplement à éjecter la cassette.

La programmation est extrêmement simple; il suffit, pour chaque séquence introduite sur le tiroir maître par les touches SQ, SQ+1, ou SQ-1, de typer sur les tiroirs esclaves les ordres qui les concernent dans la séquence entrée.

Autrement dit, pour l'exécution d'une pièce donnée, il faut étudier cette dernière comme un mécanicien le ferait pour un travail sur une machine conventionnelle et les phrases qu'il prononcerait : «Je vais d'abord amener le chariot à une certaine cote, puis percer un trou avec telle mèche, etc...» servent directement à l'élaboration du programme sans passer par un langage quelconque d'informaticien.

Le programme introduit se déroule de la façon suivante: le tiroir maître s'occupe du déroulement des séquences, mais non du contenu de celles-ci. Il part de la séquence zéro et enchaîne les suivantes par incrémentation ou par sauts suivant le jeu des JUMP et des CALL, de façon similaire à la lecture du logiciel par un ordinateur. Au début de chaque séquence, le tiroir MAITRE distribue le travail à effectuer et attend les quittances de tous les tiroirs en travail pour passer à la séquence suivante.

Les tiroirs ESCLAVES eux ne font qu'exécuter le travail qui leur est propre dans les conditions posées par le tiroir MAITRE suivant le programme introduit.

Il est évident que la puissance d'un tel dispositif est énorme, vu la capacité et la diversité de travail permises.

La description ci-dessus fait bien ressortir les avantages suivants du dispositif :

— Possibilité de composer des dispositifs de commande de complexités diverses grâce à l'utilisation d'un matériel standard (même carte selon fig. 3A et 3B) pour tous les tiroirs et surtout grâce au fait que chaque tiroir ajouté renferme lui-même le logiciel maître nécessaire à sa conversation avec le tiroir MAITRE.

— Liaison conversationnelle directe entre les tiroirs esclaves permettant d'éviter de devoir charger et décharger une mémoire générale.

— Grande facilité de programmation du fait que pour chaque séquence entrée dans le tiroir MAITRE les opérations à exécuter sont entrées dans les tiroirs ESCLAVES qui sont affectés à leur exécution.

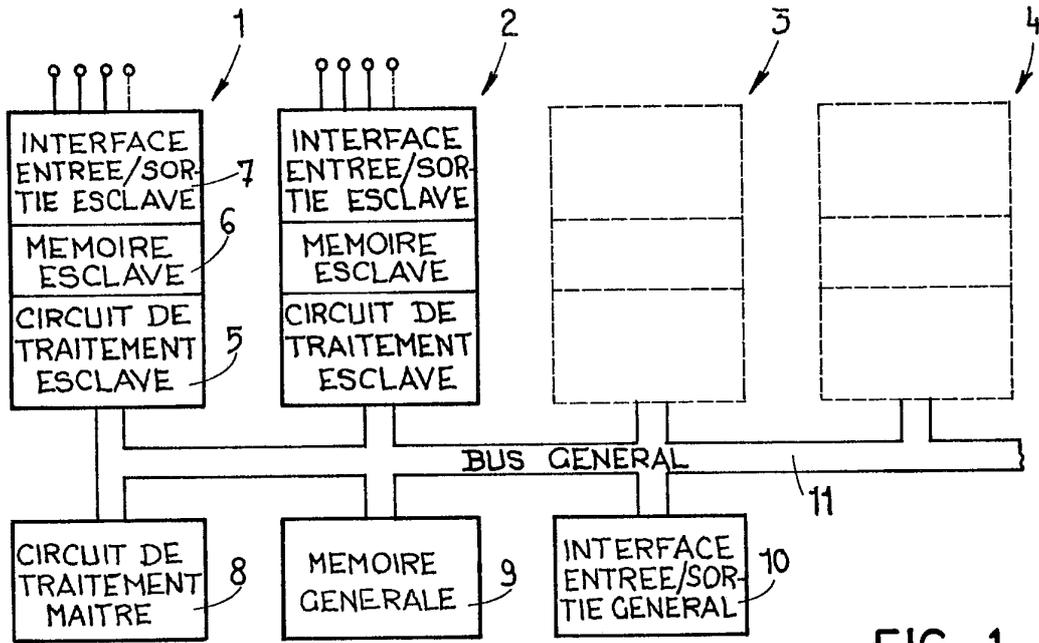


FIG. 1

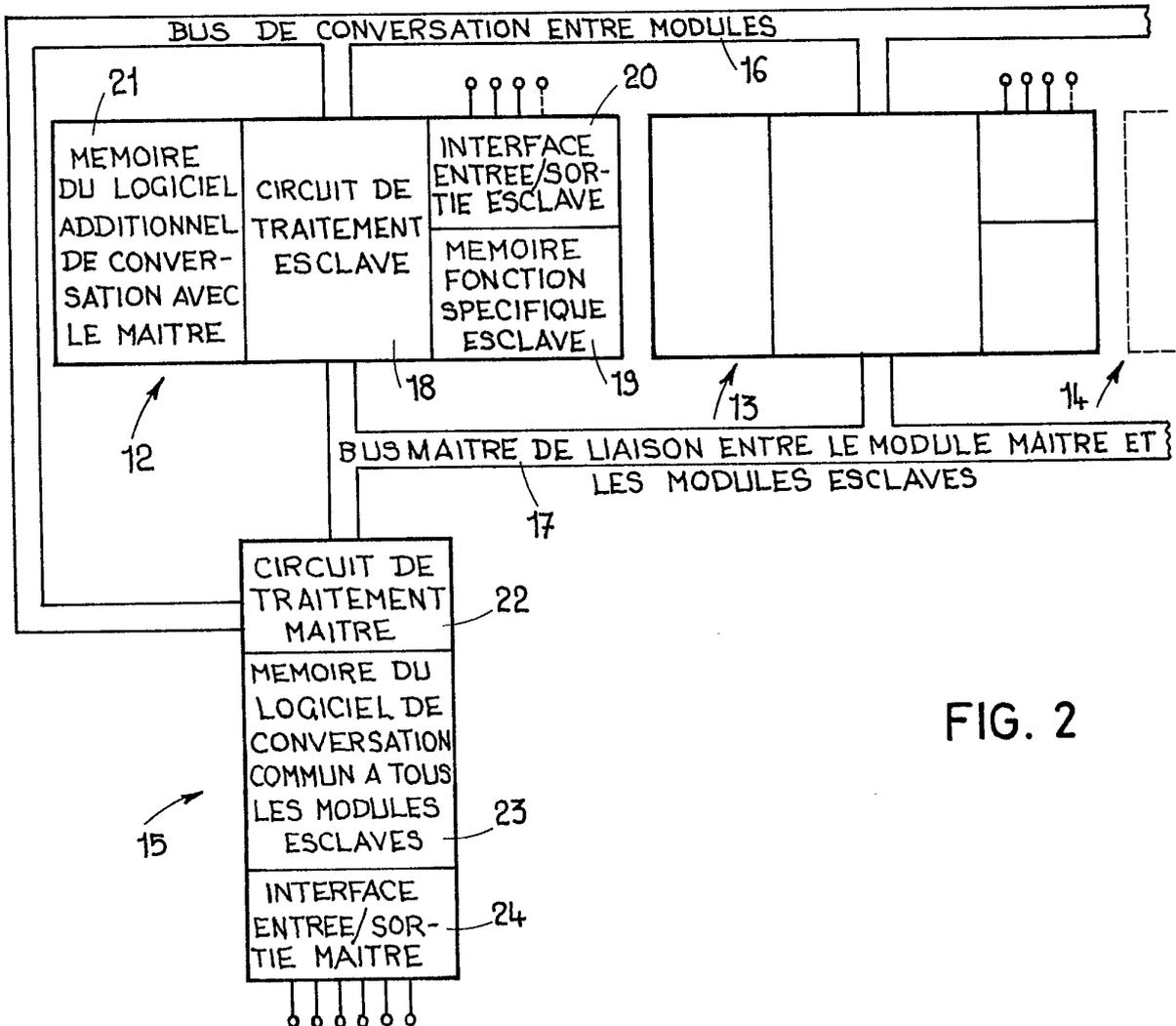


FIG. 2

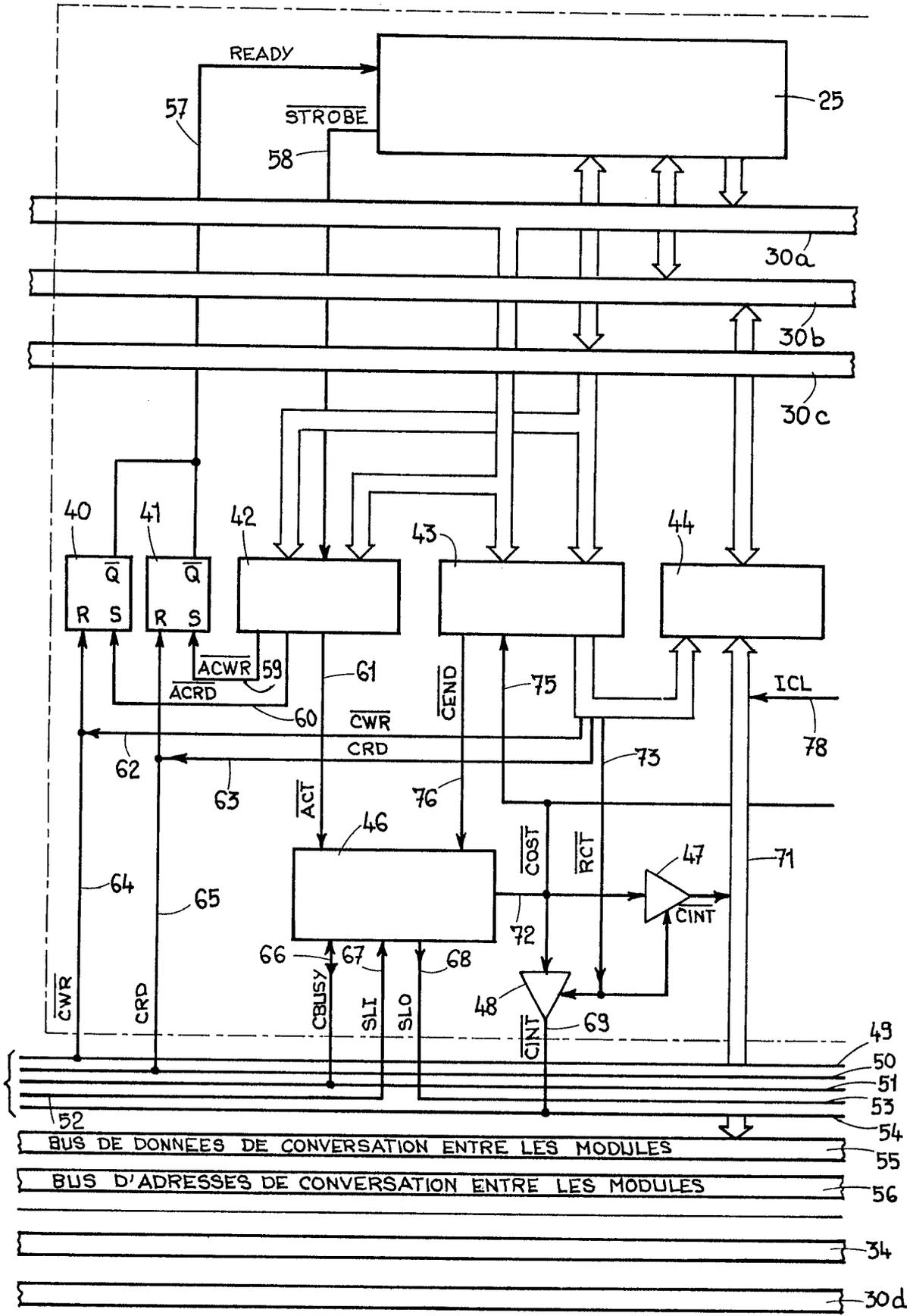


FIG. 3A

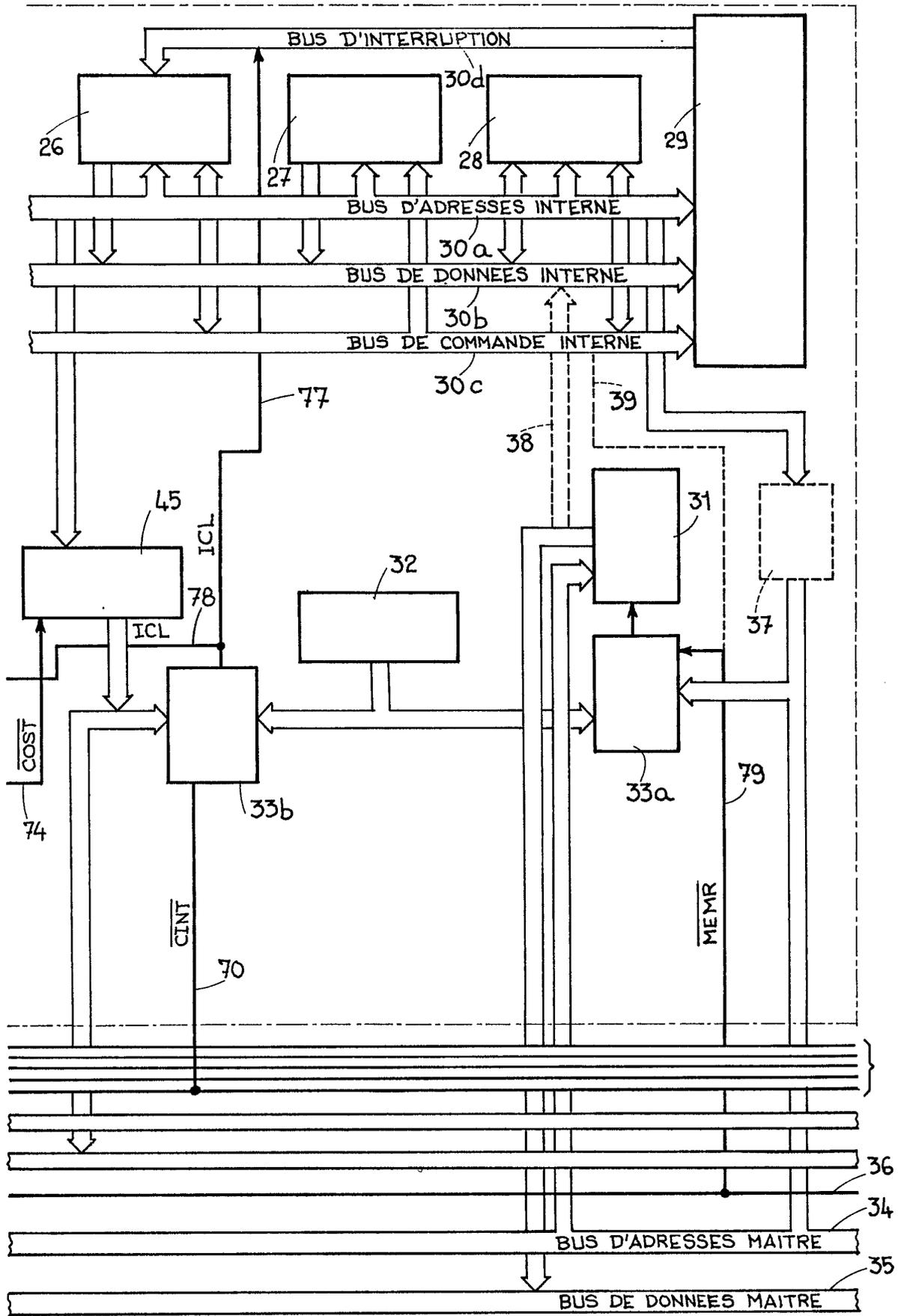


FIG. 3B

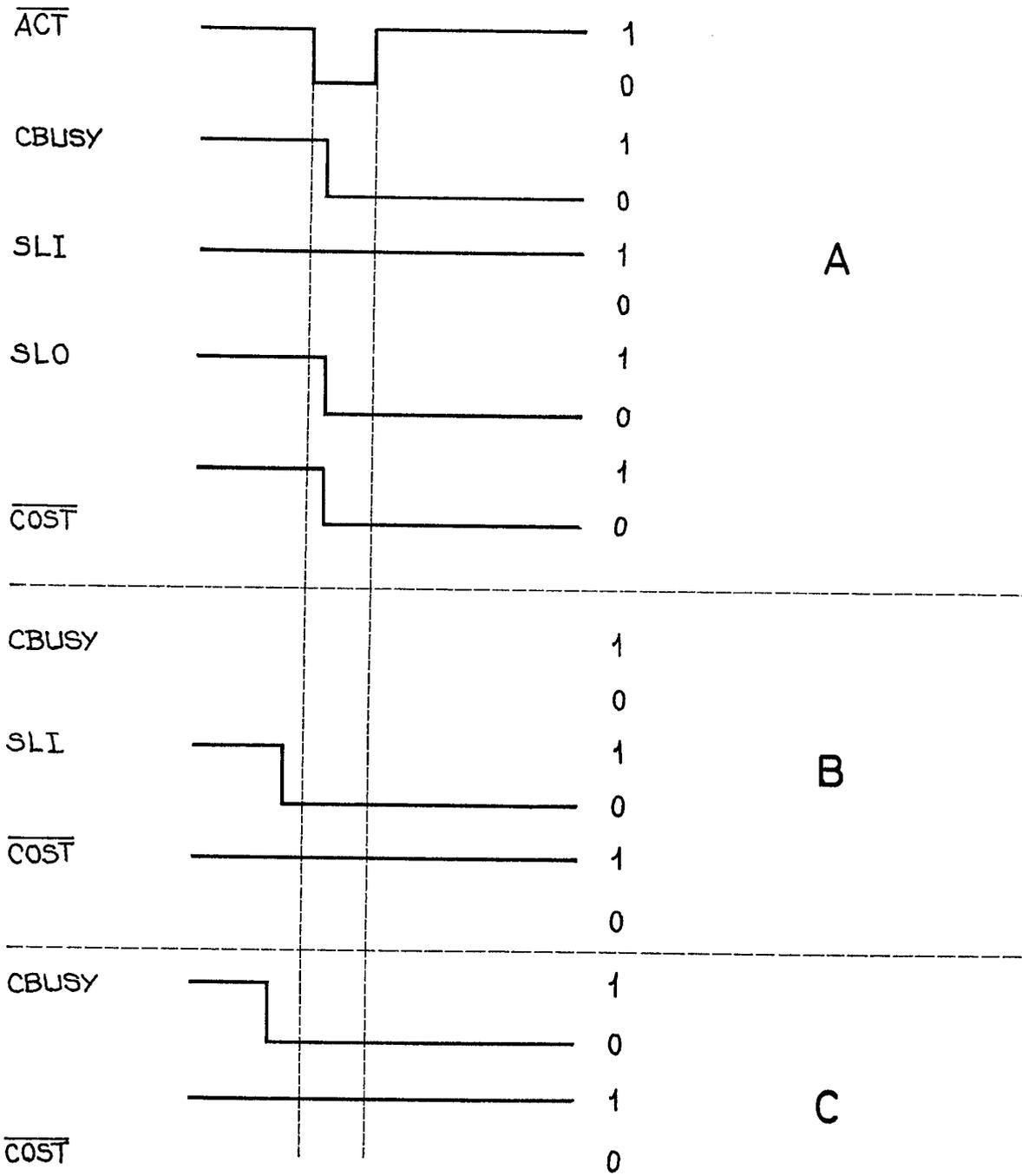


FIG. 4

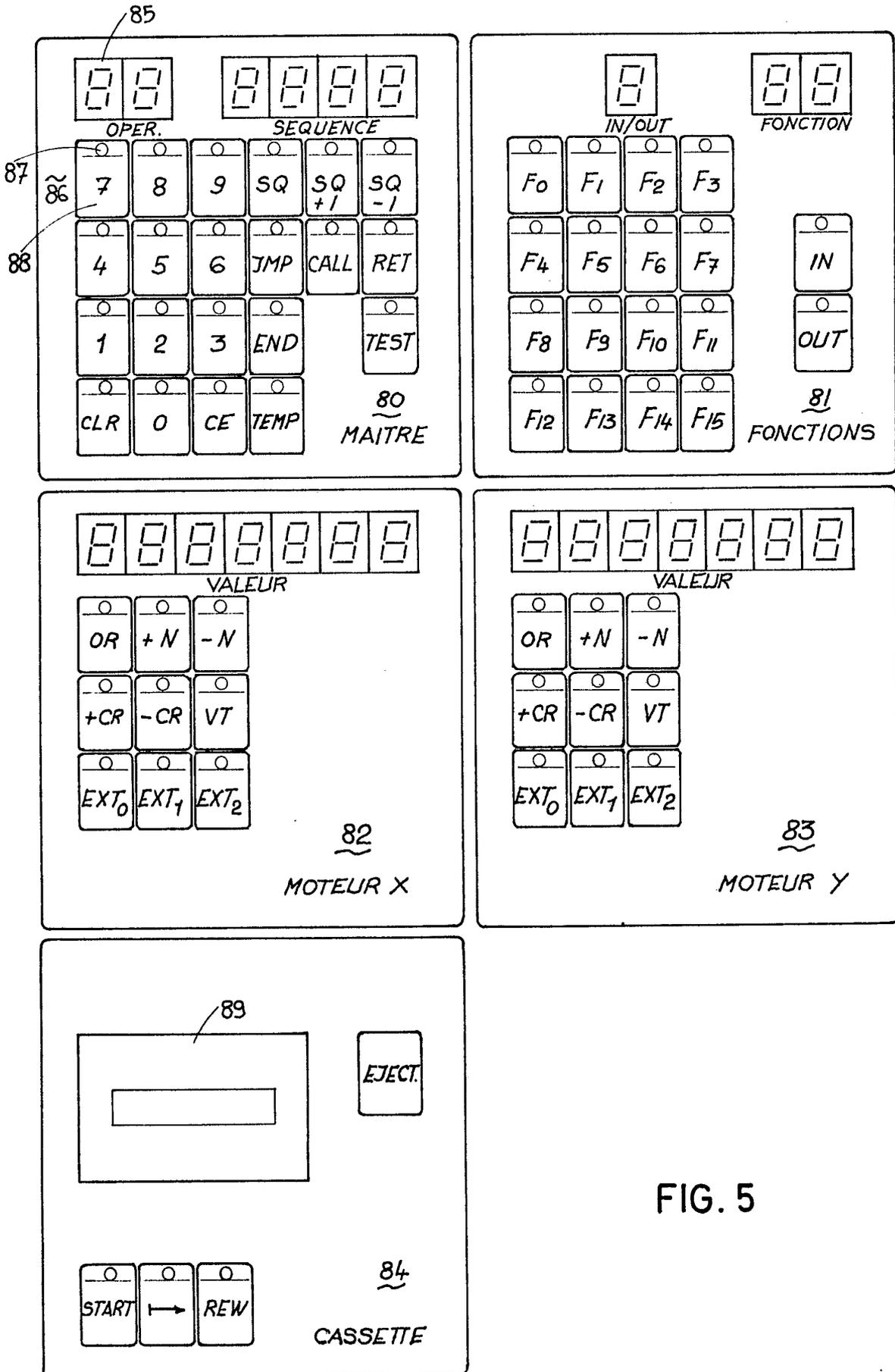


FIG. 5