



CONFÉDÉRATION SUISSE

OFFICE FÉDÉRAL DE LA PROPRIÉTÉ INTELLECTUELLE

⑤¹ Int. Cl.³: G 06 F 3/04
G 06 F 9/00
H 04 L 11/02

Brevet d'invention délivré pour la Suisse et le Liechtenstein

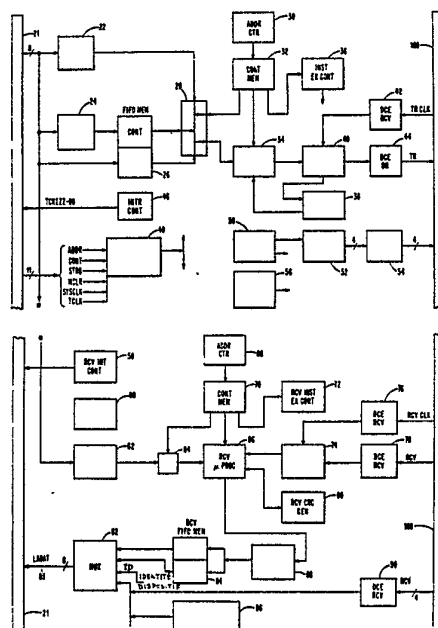
Traité sur les brevets, du 22 décembre 1978, entre la Suisse et le Liechtenstein

⑫ FASCICULE DU BREVET A5

⑪

645 470②¹ Numéro de la demande: 7215/80②² Date de dépôt: 26.09.1980③⁰ Priorité(s): 28.09.1979 US 079961④² Brevet délivré le: 28.09.1984④⁵ Fascicule du brevet
publié le: 28.09.1984⑦³ Titulaire(s):
Honeywell Information Systems Inc.,
Waltham/MA (US)⑦² Inventeur(s):
Raymond, James C., Framingham/MA (US)⑦⁴ Mandataire:
Patentanwälte Dr.-Ing. Hans A. Troesch und
Dipl.-Ing. Jacques J. Troesch, Zürich**⑤⁴ Adaptateur pour servir d'interface entre un processeur de communication et un canal de communications.**

⑤⁷ L'adaptateur comprend un dispositif logique de transmission de données et un dispositif logique de réception de données. Les dispositifs logiques de réception et de transmission comprennent des microprocesseurs (34, 66) et des mémoires de commandes à programmation câblées (32, 70) pour commander les microprocesseurs respectifs. En opération de transmission des caractères de données et de contrôle, représentant l'état des caractères de données, sont entrés dans des sections respectives de mémoire (26). Les sorties de ces sections sont transmises par multiplexeur (28) au microprocesseur (34). Le processeur décode et traduit les caractères d'information de données et de commande à son entrée, y joint des caractères d'examen de faute, pour former des cadres de caractères successifs. Ces caractères sont transmis en forme sérielle à des moyens de synchronisation (40, 44) qui les transmettent à un canal (100) de communication. En état de réception, l'information reçue est transmise dans une mémoire (74) où ces bits sont assemblés, puis transférés au microprocesseur (66). Ce dernier les examine au point de vue d'erreur et en élimine des bits d'examen d'erreur. Puis ces bits sous forme parallèle sont successivement entrés dans des mémoires (88, 84) où ils sont séparés en caractères de données et caractères d'état. Un multiplexeur (82) les transmet, ensemble avec d'autres informations, par un canal (21), vers le processeur de communication.



REVENDECATIONS

1. Adaptateur pour servir d'interface entre un processeur de communications et un canal de communications, comprenant un dispositif logique de transmission de données et un dispositif logique de réception de données, caractérisé en ce que le dispositif logique de transmission comprend

— un dispositif de registrement (22, 24) pour recevoir et stocker des caractères de contrôle transférés du processeur au début d'un cycle de transmission de données;

— une mémoire tampon (26) pour recevoir et stocker des caractères de données transférés du processeur;

— un dispositif de traitement (28, 34, 40) relié à la mémoire tampon (26) pour transmettre les caractères de données, stockés dans cette mémoire tampon (26), bit par bit en série sur le canal de communications, et

— un dispositif de contrôle (30, 32, 36) relié audit dispositif de traitement (28, 34, 40) pour commander l'opération dudit dispositif de traitement, et comprenant des moyens pour engendrer un code de contrôle de longueur de caractère, à partir des caractères de contrôle stockés dans le dispositif de registrement, et des moyens répondant audit code de contrôle de longueur de caractère pour commander le dispositif de traitement afin qu'il transmette sur le canal, pour chaque caractère transmis, un nombre de bits égal au nombre représenté par ledit code de contrôle.

2. Adaptateur selon la revendication 1, caractérisé en ce que le dispositif de contrôle (32) comprend des moyens pour engendrer un code de contrôle de longueur de caractère, à partir d'un caractère de contrôle stocké dans ledit dispositif de registrement (22), et des moyens répondant à ce code de contrôle de longueur de caractère pour commander le dispositif de traitement (28, 34, 40) afin qu'il transmette lesdits caractères de données stockés sur le canal avec une séquence de bits suivant leur ordre de priorité.

3. Adaptateur selon l'une des revendications 1 ou 2, caractérisé en ce que la mémoire tampon (26) comprend des moyens de stockage (01B06, 03B06, 05B06, 07B06) premier-entré premier-sorti.

4. Adaptateur selon l'une des revendications précédentes, dans lequel les caractères de contrôle stockés dans le dispositif de registrement (24) comprennent des caractères de commencement et de fin de cadre, qui sont transférés du processeur au commencement et à la fin d'un cycle de transmission de données respectivement, caractérisé en ce que le dispositif de contrôle (32) comprend des moyens d'engendrement (09B08, 05D08, 0K07) reliés au dispositif de registrement (22) pour engendrer un signal de contrôle indiquant que la mémoire tampon est vide si la mémoire tampon est vidée desdits caractères de données avant que le caractère de fin de cadre soit reçu, et des moyens répondant à ce signal de contrôle en transmettant des caractères de remplissage sur le canal.

5. Adaptateur selon la revendication 4, caractérisé en ce qu'il comprend un dispositif d'état (42, 82, 86) pour engendrer des signaux d'état et d'interruption pour le processeur de communications, ce dispositif d'état étant établi par le dispositif de contrôle pour indiquer une condition under-run et pour engendrer une commande d'interruption pour le processeur de communications quand un caractère fin de cadre est reçu dudit processeur.

6. Adaptateur selon l'une des revendications 4 ou 5, caractérisé en ce que les caractères de remplissage comprennent une séquence d'abandon suivie d'une série de délimiteurs.

7. Adaptateur selon l'une des revendications précédentes, caractérisé en ce que le dispositif logique de réception comprend un dispositif de registrement (62) pour recevoir et stocker un caractère de contrôle, transféré du processeur au commencement d'un cycle de réception de données, un dispositif de traitement (66, 74, 80) pour recevoir des données du canal bit par bit en série et pour l'assemblage de ces bits reçus sous forme de caractères de données à bits parallèles, un dispositif (84, 82, 88) de transfert pour transférer lesdits caractères de données à bits parallèles au processeur de communications, et un dispositif de contrôle (68, 70, 72) relié audit dis-

positif de traitement pour contrôler ses opérations, le dispositif de contrôle comprenant des moyens d'engendrement pour engendrer un code de contrôle de longueur de caractère, à partir d'un caractère de contrôle stocké dans le dispositif de registrement, et des moyens répondant à ce code de contrôle de longueur de caractère pour commander le dispositif de traitement afin que les bits de données reçus soient assemblés dans des caractères de données à bits parallèles avec un nombre de bits équivalant au nombre représenté par ledit code de contrôle.

8. Adaptateur selon la revendication 7, caractérisé en ce que le dispositif de transfert comprend: une mémoire tampon premier-entré premier-sorti (84) pour recevoir et stocker des caractères de données assemblés en bits parallèles dudit dispositif de traitement, des moyens de chargement pour charger un caractère d'état dans la mémoire tampon avec chacun desdits caractères de données, et un multiplexeur (82) pour transférer les caractères de données et d'état de la mémoire tampon au processeur de communications, ces moyens de chargement de caractère d'état comprenant un moyen d'entrée pour entrer un code fin de cadre dans le caractère d'état qui accompagne le dernier caractère de données reçu.

9. Adaptateur selon la revendication 8, caractérisé en ce que le dispositif de contrôle (70) comprend des moyens d'engendrement pour engendrer un signal de contrôle indiquant que la mémoire tampon est pleine si l'espace de stockage accessible dans la mémoire tampon 84 est plein avant que ledit dernier caractère de données y soit entré, et des moyens répondant à ce signal de contrôle pour commander le dispositif de traitement afin qu'il entre un code over-run dans le caractère d'état qui accompagne ce dernier caractère de données, fournissant une indication d'un état over-run au processeur de communications.

10. Adaptateur selon l'une des revendications 1 à 9, caractérisé en ce qu'il comprend des moyens de chargement pour charger un caractère d'état dans la mémoire tampon avec chacun des caractères de données, le dispositif de contrôle comprenant des moyens qui contrôlent le dispositif de traitement afin de déterminer le nombre de bits dans un dernier de ces caractères de données et pour entrer un code de contrôle de longueur de ce dernier caractère dans le caractère d'état qui accompagne ce dernier caractère de données, ce code de contrôle désignant le nombre de bits dans ce dernier caractère.

11. Adaptateur selon la revendication 10, caractérisé en ce que le dispositif de traitement est commandé par le dispositif de contrôle afin d'entrer une partie d'un code de contrôle du dernier caractère dans le caractère d'état qui accompagne ledit dernier caractère de données, si le nombre de bits dans ce dernier caractère de données est inférieur au nombre de bits dans les caractères de données précédents.

12. Adaptateur selon l'une des revendications précédentes, caractérisé en ce que la mémoire tampon (26) recevant des caractères de données du processeur pour leur transmission sur le canal de communications comprend une partie reliée au dispositif de registrement pour recevoir des caractères de contrôle pour cette réception et est mise en œuvre de manière que chaque cadre de caractères de données apporté au dispositif de traitement inclut un champ de texte et un champ de contrôle et en ce que le dispositif de contrôle comprend des moyens d'engendrement qui engendrent un code de contrôle de grandeur de champ à partir d'un caractère de contrôle stocké, ce code de contrôle indiquant au dispositif de traitement la grandeur du champ de contrôle dans le cadre.

13. Adaptateur selon l'une des revendications précédentes, caractérisé en ce qu'il comprend un dispositif (38) qui exécute un contrôle de faute sur les caractères de données quand ils sont transmis sur le canal de communications, et que le dispositif de contrôle comprend des moyens qui commandent le dispositif de traitement afin qu'il transmette un caractère de séquence de contrôle sur ce canal suivant le dernier caractère de données d'un cadre de caractères de données transmis.

14. Adaptateur selon les revendications 4 et 13, caractérisé en ce que le dispositif de contrôle comprend des moyens de transfert pour

transférer le caractère de fin de cadre du processeur avec le dernier caractère de données et des moyens d'engendrement pour engendrer un code de contrôle d'inhibition du contrôle de faute, en réponse à ce caractère de fin de cadre, le dispositif de traitement répondant à ce code de contrôle d'inhibition en permettant à ces caractères d'être transmis sans ledit contrôle de faute.

La présente invention se rapporte à un adaptateur pour servir d'interface entre un processeur de communication et un canal de communications, comprenant un dispositif logique de transmission de données et un dispositif logique de réception de données.

Des adaptateurs matériels ont été utilisés dans les systèmes de communications afin de réaliser le transfert d'informations entre une ligne de transmission et un ensemble de traitement de données. Des modifications intervenues dans le protocole des communications ont suscité de nombreuses demandes d'adaptateurs. Ces demandes ont été satisfaites par un accroissement des matériels, ce qui a conduit à une augmentation des coûts de fabrication et des dimensions physiques des équipements.

Par essence, une communication consiste en un transfert d'informations bidirectionnel simultané. Les adaptateurs matériels nécessitent deux systèmes séparés, un pour les opérations de réception et un pour les opérations de transmission. Il est aussi reconnu que les messages peuvent avoir des longueurs variables et peuvent comprendre des multiplets de données de longueurs variables. Les réalisations matérielles assurent une souplesse fort limitée, au prix d'un coût élevé et d'un encombrement accru.

Les problèmes associés aux adaptateurs de communications matériels ont été partiellement surmontés par la venue de contrôleurs de communications comprenant du matériel et une certaine forme d'ensemble de commande à programmation câblée. Bien que la duplication de systèmes entiers ait été évitée pour réaliser les opérations de transmission et de réception, la quantité de matériels utilisés est toujours importante par suite d'un mariage inefficace des fonctions de la programmation câblée et de celles des matériels. Il en est résulté un certain compromis en ce qui concerne les vitesses de transfert d'informations que peut assurer le système.

En outre, la capacité en traitement de données des contrôleurs de communications connus est relativement limitée, de sorte que la réalisation de liaisons de communications large bande a jusqu'à présent été difficile, voire impossible.

L'invention vise à éliminer les problèmes associés aux adaptateurs connus. Ce but est atteint par un adaptateur tel que spécifié plus haut, qui est caractérisé selon la revendication 1.

L'invention est exposée plus en détail dans ce qui suit, sur un mode de réalisation préféré illustré aux dessins ci-annexés sur lesquels:

les fig. 1 à 12 sont douze feuilles de circuits logiques qui constituent ensemble le circuit de l'adaptateur de ligne de communications selon l'invention: chaque figure se compose de deux parties (par exemple fig. 1a et fig. 1b) qui doivent être réunies côte à côte, la partie a étant disposée à la gauche de la partie b;

la fig. 1 illustre les connecteurs d'interface entre le processeur de communications multiligne (MLCP) et l'équipement de communications de données (DCE);

la fig. 2 représente de façon générale la logique de décodage d'adresse et de contrôle;

la fig. 3 représente la logique de la commande d'horloge de système et d'essai,

la fig. 4 représente la logique des multiplexeurs d'entrée et de test du processeur MLCP;

la fig. 5 représente les circuits de commande et de récepteurs de l'équipement de communications de données DCE,

la fig. 6 représente la mémoire FIFO (premier-entré premier-sorti) de transmission et la logique de commande;

la fig. 7 représente le compteur d'adresse de transmission et la logique de mémoire de commande de transmission;

la fig. 8 représente le microprocesseur de transmission et la logique de commande d'instructions associée;

la fig. 9 représente le compteur d'adresse de réception et la logique de mémoire de commande de réception;

la fig. 10 représente le microprocesseur de réception et la logique associée;

la fig. 11 représente la logique de commande d'instructions de réception;

la fig. 12 représente la mémoire FIFO de réception et la logique de commande associée;

la fig. 13 est un schéma synoptique d'un système au sein duquel peut fonctionner l'invention,

la fig. 14 est un schéma synoptique de l'adaptateur de ligne selon l'invention: la figure se compose de deux parties, fig. 14a et fig. 14b, devant être assemblées, la partie a au-dessus de la partie b;

les fig. 15a-g illustrent le format de différents mots de commande et d'état utilisés dans l'adaptateur selon l'invention;

la fig. 15h illustre le format de colonne utilisé pour les messages dans la commande de transmission de données à haut niveau;

la fig. 16 est un schéma simplifié du microprocesseur de transmission 4 bits;

la fig. 17 représente le microprocesseur de réception, montrant la manière dont deux des microprocesseurs de la fig. 16 sont interconnectés pour constituer un microprocesseur 8 bits destiné à traiter les données reçues;

la fig. 18 montre les fonctions de base exécutées par le microprogramme de transmission;

la fig. 19 indique les fonctions de base exécutées par le microprogramme de réception;

les fig. 20a, b et c forment une table des sources de signaux, dans laquelle chaque signal utilisé dans l'adaptateur selon l'invention est repris dans l'ordre alphabétique en même temps qu'une désignation de l'élément du circuit qui est à l'origine de ce signal.

On se réfère d'abord à la fig. 13 qui montre un schéma synoptique d'un ensemble ordinateur comprenant un processeur de communications multiligne 10 en communication électrique avec un processeur central 11 et une unité mémoire principale 12, par l'intermédiaire d'un bus de communications commun qui sera appelé dans la suite mégabus 13.

Ce mégabus constitue une voie de communication d'informations entre deux unités de l'ensemble. Les trajets d'information sont asynchrones par conception, ce qui permet à des unités travaillant à des vitesses différentes de fonctionner de manière efficace. Le mégabus assure le transfert d'informations, y compris les demandes de communications, les ordres de commande, les signaux d'état et les transferts de données entre l'unité mémoire 12 et le processeur de communications 10.

Une unité quelconque de l'ensemble demandant une communication avec un autre ensemble lance une demande de cycle de bus. Lorsque le cycle de bus est accordé, l'unité-demanderesse devient l'unité maîtresse, tandis que l'unité adressée devient l'unité esclave. Certains échanges sur le bus requièrent un cycle de réponse ainsi qu'un cycle de demande. Par exemple, l'unité maîtresse peut s'identifier elle-même à une unité esclave et indiquer qu'une réponse est requise. Lorsque l'information requise devient disponible, l'unité esclave assume le rôle d'unité maîtresse et lance un transfert d'information vers l'unité demanderesse. Une description plus détaillée de l'ensemble illustré à la fig. 13 est donnée dans le brevet des Etats-Unis d'Amérique N° 3993981 dont la titulaire est l'ayant droit.

Lors du service des demandes de cycle de bus, le processeur central CPU 11 a la priorité la plus faible, le processeur de communications MLCP 10 a la plus faible priorité suivante, et l'unité mémoire MEM 12 a la priorité la plus élevée. Le processeur de communications 10 est également en communication avec un adaptateur

de ligne de communications de commande de transmission de données à haut niveau à large bande BHCLA 14, et avec d'autres adaptateurs de lignes de communications 15. L'adaptateur 14 peut à son tour être connecté directement à un équipement de traitement de données tel qu'un processeur central ou tout autre type d'équipement terminal de données DTE. En mode télé-informatique, l'adaptateur de ligne 14 peut être connecté, par l'intermédiaire d'une unité de communications de données locale DCE 16 et d'un câble de communications 17, à un équipement de communications de données 18 faisant partie d'un réseau de communications de données DCN. L'équipement de communications de données distant 18 peut, à son tour, être connecté à un terminal de données distant 19. Les équipements de communications de données DCE sont capables de moduler et démoduler des flots d'informations binaires et peuvent donc recevoir et transmettre des informations.

L'adaptateur de ligne 14 peut être rendu compatible avec n'importe quel type d'interface d'équipement de communications de données conforme aux ensembles de données de types bien connus EIA RS-232-C, MIL 18-C ou EIA RS-XYZ, CCITT V. 35 ou BELL 301 ou 303. La forme de réalisation préférée de l'invention décrite dans ce qui suit est compatible avec les unités d'interface type BELL 301 ou 303.

Le processeur de communications 10 avec lequel la forme de réalisation préférée de l'invention est destinée à fonctionner est un type connu de processeur de communications multiligne fabriqué par Honeywell Information Systems Inc. et décrit de manière générale dans la demande de brevet E.U.A. N° 911635, déposée le 1^{er} juin 1978 aux noms de James C. Raymond, R.A. Lemay et R.P. Kelly. Le fonctionnement du processeur de communications se trouve décrit dans le «Manuel de programmation à l'usage de l'utilisateur», Honeywell Series 60 (Level 6) Communications Handbook, Order No. AT 97 REV. 2, octobre 1978. L'adaptateur de ligne de communications selon l'invention est représenté sous forme synoptique à la fig. 14 complétée par les schémas de bloc logiques montrés aux fig. 1 à 12.

La structure et le fonctionnement de l'adaptateur de ligne selon l'invention sont décrits ci-après avec référence à la fig. 14, qui comporte deux feuilles, fig. 14a et fig. 14b. L'adaptateur de ligne comprend deux connecteurs d'interface 21 et 100 qui se trouvent connectés, respectivement, au processeur de communications 10 et à l'équipement de communications de données DCE local 16 (fig. 13). Fondamentalement, la fig. 14a représente les circuits de transmission de l'adaptateur de ligne et la fig. 14b représente les circuits de réception.

Le processeur de communications 10 exerce le contrôle du système par l'intermédiaire d'un ensemble de circuits logiques 48 qui reçoivent du processeur de communications 10 des signaux d'adresse ADDR, des signaux de commande CONT, un signal de pointage STRB, un signal remise à zéro de l'unité maîtresse MCLR, et des signaux d'horloge SYSCLK et TCLK par l'intermédiaire du connecteur 21. En réponse à ces signaux, les circuits logiques 48 produisent un ensemble de signaux de commande et de temps pour les autres circuits du système de transmission et de réception, afin d'effectuer les fonctions de synchronisation et de commande nécessaires. Les circuits de transmission comprennent un registre de mots de commande de transmission 24 et un registre de mots de commande étendu de transmission 22. Ces registres reçoivent du processeur de communications 10 des mots de commande comptant, respectivement, 8 et 4 bits. Le format du mot de commande étendu de transmission est montré à la fig. 15a, et les deux formats du mot de commande de transmission sont montrés à la fig. 15b (format début de colonne) et à la fig. 15c (format fin de colonne).

Une mémoire tampon premier-entré premier-sorti FIFO de transmission 26 sert à stocker temporairement les données de commande de message reçu du processeur de communications 10. Le fonctionnement des circuits de la mémoire 26 est tel que les multiplets de données et de commande reçus soient rapidement acheminés entre l'entrée et la sortie pour former une file d'attente de multiplets de données et de commande. Ces multiplets sont transférés au mi-

croprocesseur de transmission 34 par l'intermédiaire du multiplexeur 28 et du microprocesseur 34; ces multiplets sont appliqués sur le canal de communications, bits en série, par l'intermédiaire du connecteur d'interface 100. La commande de ce fonctionnement est effectuée par une programmation câblée, stockée dans la mémoire de commande 32. Un compteur d'adresse 30 et un circuit de commande d'exécution d'instructions 36 commandent l'exécution de la programmation câblée afin de réaliser les opérations de commande de transmission appropriées. Le flot de données transmises est canalisé vers le connecteur 100 par un circuit de synchronisation de bits de données 40 et un circuit de commande d'équipement de communications de données 42. Un signal d'horloge de transmission obtenu de l'équipement de communications de données DCE est reçu par le circuit récepteur 42 et appliqué au circuit de synchronisation 40 afin de synchroniser le flot de données vers le canal de communications. Un générateur de signaux de parité cycliques 38 surveille le flot de bits et envoie au microprocesseur 34 des données de contrôle de parité cycliques afin de corriger les erreurs.

Des informations de commande sont également appliquées au canal de communications par l'intermédiaire du registre de mot de commande 50, du registre mémoire secondaire 52, et des circuits de commande d'équipement de communications de données 4 bits 54. Un circuit de commande d'interruption 46 délivre un signal d'interruption TCRIZZ au processeur de communications 10, par l'intermédiaire du connecteur 21, afin de commander les interruptions et réponses appropriées. Des circuits logiques de mode d'essai 56 sont également prévus, mais ils ne font pas partie essentielle de l'adaptateur selon l'invention et ne seront donc pas décrits ici.

Les circuits de réception de l'adaptateur de ligne (fig. 14b) comprennent des récepteurs 76, 78, 90 pour recevoir de l'équipement de communications de données, par l'intermédiaire du connecteur 100, des signaux d'horloge de réception, des signaux binaires de données de réception et un signal d'état de réception 4 bits. Les bits de données sont transmis du récepteur 78 au circuit de synchronisation 74, et ils sont transférés de celui-ci au microprocesseur de réception 66. Celui-ci fonctionne sous la commande de la programmation câblée de réception stockée dans la mémoire de commande 70 afin d'assembler les bits de données en multiplets parallèles, afin d'exécuter le contrôle d'erreur de parité par l'intermédiaire du générateur 80 et afin d'exécuter une variété de fonctions additionnelles (surveillance, mise en format et commande). Un compteur d'adresse de réception 68 et un circuit de commande d'exécution d'instructions 72 assurent le contrôle de l'exécution de la programmation câblée.

Le processeur de communications 10 fournit l'information de commande par l'intermédiaire d'un mot de commande de réception qui se trouve stocké dans le registre de commande de réception 62 et appliqué au microprocesseur 66, par l'intermédiaire d'un multiplexeur 64. Un circuit de commande d'interruption de réception 58 assure le transfert des ordres d'interruption et de réponse provenant des circuits de réception vers le processeur de communications 10, par l'intermédiaire du connecteur 21.

Les multiplets de données assemblés sont transférés du microprocesseur 66 dans un circuit tampon 68 d'où ils sont canalisés vers les circuits de la mémoire FIFO de réception 84. Cette mémoire, tout comme la mémoire 26 des circuits de transmission, comprend une section pour les données de message et une section pour les données d'état. La mémoire 84 achemine les multiplets reçus de manière rapide afin de créer automatiquement une file d'attente de multiplets destinés à être transférés au processeur de communications 10 sous forme de signaux LADAT, par l'intermédiaire du multiplexeur 82 et du connecteur 21.

Le multiplexeur 82 sert également à autoriser le processeur de communications 10 à lire l'identité d'un dispositif ID, lorsqu'il reçoit la première fois un ordre d'interruption de récepteur, et à lire le signal d'état de réception fourni par l'équipement de communications de données DCE par l'intermédiaire des récepteurs 90. En outre, des signaux de contrôle adaptateurs prêts under-run, engendrés par le circuit de commande 86, sont également envoyés au pro-

cesseur de communications 10 par l'intermédiaire du multiplexeur 82. Un circuit de resynchronisation de récepteur 60, fonctionnant sous la commande de la programmation câblée de transmission ou de réception, sert à resynchroniser automatiquement les circuits récepteurs avec le canal de communications en vue de la réception de données. Cette fonction conditionne préalablement les circuits récepteurs de telle manière que le processeur de communications 10 n'ait pas à répondre aussi rapidement à un ordre d'interruption de réception.

D'une façon générale, les circuits de transmission de l'adaptateur de ligne de communications fonctionnent de la manière suivante. Le processeur de communications 10 initialise le système en actionnant la logique de décodage 48 au début d'une opération de transmission. Le processeur 10 sélectionne un adaptateur particulier et règle les registres appropriés par l'intermédiaire des lignes d'adresse et de commande connectées à la logique de décodage 48. L'information d'identification engendrée par l'identification ID de dispositif est transmise au processeur de communications, par l'intermédiaire du multiplexeur 82, afin d'indiquer l'identité de l'adaptateur qui répond. Le processeur de communications 10 transfère alors les bits de commande à l'équipement de communications de données DCE, par l'intermédiaire du registre 50, afin de fournir une réponse appropriée à l'équipement DCE. Ensuite, le processeur de communications 10 lit l'état de l'équipement DCE, par l'intermédiaire des récepteurs 90, et le multiplexeur 82 afin de déterminer si cet équipement DCE est prêt à recevoir les données transmises.

Ensuite, le processeur de communications 10 charge les registres 22 et 24, puis il commence à transférer les multiplets de données et de commande dans la mémoire 26. Le processeur de communications continue à charger la mémoire avec les données transmises jusqu'à ce que cette mémoire soit pleine, ce qui est indiqué par le signal TAR engendré par le circuit 86. Lorsque la mémoire est ainsi remplie, le circuit de commande d'interruption 46 informe le processeur de communications 10 qu'il a à cesser d'envoyer des données.

Dès que l'équipement DCE est prêt à recevoir les données, le microprogramme de transmission dans la mémoire de commande 32 actionne les circuits de transmission pour prendre les multiplets de données de commande à la sortie de la mémoire 26 et les transmettre sur le canal de communications.

Pendant cette opération, les données sont transmises colonne par colonne, bits en série, conformément au format montré à la fig. 15h. Le programme de transmission câblé, qui commande par l'intermédiaire du microprocesseur 34, surveille la séquence des colonnes et commande le système en conséquence. Au début de chaque colonne, le microprogramme commande une vérification pour voir si une donnée valable est présente à la sortie de la mémoire 26 et, dans l'affirmative, une séquence de code complémentaire initiale est transmise et le générateur de contrôle de parité cyclique 38 se trouve initialisé.

Si une donnée valable n'est pas présente à la sortie de la mémoire 26, le microprogramme commande la transmission de séquences code complémentaire ou bit abandon, comme spécifié par le bit TIFM stocké dans le registre 22 (fig. 15a). Pendant la transmission d'une colonne de message, le microprogramme commande le transfert de multiplets de la mémoire 26 dans le microprocesseur 34 et l'envoi de données par le microprocesseur, bit par bit, sur le canal de communications, par l'intermédiaire du circuit de synchronisation 40 et du circuit de commande 44. A la fin de chaque intervalle de multiplet, un nouvel ensemble de caractères de données et de commande est introduit dans le microprocesseur à partir de la mémoire 26. Un compteur longueur de multiplet, situé dans la mémoire à accès sélectif RAM du microprocesseur, se trouve rechargé avec le code de longueur de multiplet développé à partir des bits TBS dans le mot de commande de transmission (fig. 15b), afin d'autoriser le microprocesseur à commander la longueur des multiplets d'après le code TBS.

Lorsque le système détecte un bit TEOF à l'état 1 dans le multiplet de commande transféré par la mémoire 26 (fig. 15c), il lance la

séquence fin de colonne transmission. Le caractère suivant transmis sur le canal est connu comme étant le dernier caractère de la zone texte de la colonne. Après que ce caractère a été transmis, la première moitié de 8 bits du caractère de contrôle de parité cyclique est envoyée par le générateur 38, complétée et transmise sur le canal. Ensuite, la seconde moitié de 8 bits du caractère de contrôle de parité est transmise, afin de compléter la zone FCS.

Le programme commande ensuite le système pour transmettre une séquence code complémentaire de fermeture pendant l'intervalle de multiplet suivant, afin d'achever la transmission de la colonne. Le circuit de commande d'interruption 46 est établi pour envoyer un signal d'interruption au processeur de communications 10, et les registres de mots de commande 22 et 24 sont remis à l'état initial. En réponse au signal d'interruption, le processeur de communications 10 lit l'état de l'équipement DCE, par l'intermédiaire du multiplexeur 82, et il examine le bit TU dans le mot d'état DCE (fig. 15g), afin de déterminer si un état under-run est apparu. Si le bit TU est à l'état zéro, aucun état under-run n'est indiqué.

Si, pendant l'intervalle de transmission de colonne, des multiplets de données valides n'ont pas été trouvés à la sortie de la mémoire 26, un état under-run est déclaré et le microprocesseur envoie une séquence d'abandon suivie d'une série de séquences de code complémentaire. Lorsqu'une donnée valide est détectée ultérieurement à la sortie de la mémoire 26, elle est négligée jusqu'à ce que le bit TEOB à l'état un soit détecté dans le mot de commande de transmission, ce qui a pour effet de mettre à l'état un le bit TU dans le mot d'état DCE, tandis que le circuit de commande d'interruption 46 engendre un signal d'interruption pour le processeur de communications 10.

En mode réception, le fonctionnement général du système est le suivant. Le processeur de communications 10 charge un mot de commande (fig. 15d) dans le registre 62, et la logique de décodage 48 lance un ordre de resynchronisation afin de déclencher le circuit de resynchronisation 60 en vue d'initialiser les circuits de réception. La mémoire 26 est alors vidée et le système est prêt pour un cycle de réception.

Pendant chaque transition négative du signal d'horloge au récepteur 76, un bit de données au récepteur 78 se trouve copié dans le circuit tampon de la logique de synchronisation de bit de données 74. Ensuite, le programme câblé de réception commande l'introduction du bit de données dans le microprocesseur et la vérification de la présence d'une séquence de code complémentaire 01111110. Lorsque le code est détecté, le programme commande la recherche du premier caractère subséquent qui n'est pas un code complémentaire, lequel caractère est déterminé comme étant le premier caractère de la colonne de message reçu. A ce moment, le système met le générateur 80 dans l'état pour lequel il produit uniquement des bits un, afin de le préparer à engendrer le caractère de contrôle de parité cyclique, et il lance un sous-programme de commande pour commencer à assembler les caractères de réception.

Après un retard de 32 bits, le programme de réception commande le lancement du sous-programme de transfert de données par lequel un caractère assemblé se trouve chargé en parallèle par le tampon 88. Le programme commande alors le transfert du caractère dans la section données de la mémoire 84, il remet le tampon 88 à l'état initial et il commande le transfert d'un caractère d'état contenant uniquement des bits zéro du tampon 88 dans la section état de la mémoire 84.

Un nouveau point d'entrée du sous-programme de registre à décalage commande l'assemblage du caractère reçu suivant, puis la répétition du sous-programme de commande de transfert de données, afin de charger le caractère dans le tampon 88 puis dans la mémoire 84 en même temps qu'un multiplet d'état contenant uniquement des bits zéro.

Cette série d'opérations se poursuit jusqu'à ce que soit détectée une séquence code complémentaire indiquant la fin de la colonne reçue. Le programme commande alors le déroulement du sous-programme de commande courant par lequel le dernier caractère

reçu se trouve cadré à droite et vérifié quant à l'état multiplet partiel. Le programme commande la génération du mot d'état associé au dernier caractère et extrait de ce mot l'information relative à la longueur de multiplet et la longueur de multiplet partiel, afin de déterminer la présence de l'état multiplet partiel. Ensuite, le caractère de contrôle de parité cyclique cumulé est lu dans le générateur 80, 8 bits à la fois, et un contrôle de reste est exécuté. Si la valeur du reste est incorrecte, le bit RFCSE du mot d'état de réception est mis à l'état un en même temps que le bit REOF, afin d'indiquer un état d'erreur dans la zone FCS, et afin d'indiquer que la fin de colonne a été atteinte. Ensuite, les derniers multiplets de données d'état sont chargés dans la mémoire 84. Le programme commande alors l'achèvement de la séquence de colonne en transférant la zone FCS (deux multiplets de 8 bits) reçue de l'équipement DCE dans la mémoire 84 (si le mot de commande le spécifie).

Si, avant qu'un caractère reçu ne soit chargé dans la mémoire 84, il apparaît une indication selon laquelle cette mémoire est pleine, un état de surcharge se trouve indiqué. En réponse à cet état, l'adaptateur de ligne de communications traite le restant de la colonne en négligeant les caractères restants. Lorsque le bit fin de colonne se trouve détecté, un mot d'état se trouve chargé dans la mémoire 84, ce mot de commande contenant les bits RO et REOF à l'état un afin d'indiquer les états de surcharge et de fin de colonne.

Si une colonne reçue se termine par une séquence d'abandon (8 bits un), le caractère de contrôle de parité n'est pas inspecté et le bit RA de mot d'état réception est mis à l'état un en même temps que le bit REOF, afin d'indiquer la situation au processeur de communications 10.

Chaque fois que des caractères de données et d'état sont disponibles à la sortie de la mémoire 84, la commande d'interruption 58 est mise à l'état actif afin d'engendrer un ordre d'interruption pour le processeur de communications 10. Celui-ci répond, par l'intermédiaire des circuits de commande de décodage 48, en lisant les mots d'état et de données par l'intermédiaire du multiplexeur 82. Chaque opération de lecture de ce genre décale la mémoire 84 pour rendre disponibles les caractères d'état et de données suivants. Le processeur de communications 10 lit alors l'état de l'équipement DCE par l'intermédiaire du récepteur 90 et inspecte la sortie du circuit de commande 86. Si ce dernier présente un bit RAR à l'état un, le processeur 10 lit une autre paire de caractères de données et d'état dans la mémoire 84. Le processeur poursuit le cycle aussi longtemps que le bit RAR reste à l'état un. Lorsque ce bit change d'état, la commande d'interruption 58 se trouve remise à l'état initial et le processeur de communications 10 cesse de lire des données dans la mémoire 84. Chaque fois qu'un autre caractère apparaît à la sortie de la mémoire, le bit RAR passe de nouveau à l'état un et la commande d'interruption 58 engendre un autre signal d'interruption afin de reprendre le transfert vers le processeur de communications des caractères de données et d'état reçus.

Lorsque le bit REOF dans le mot d'état de réception est à l'état un, indiquant ainsi une fin de colonne, le processeur de communications 10 sait que le dernier caractère de la zone texte de la colonne apparaît à la sortie de la mémoire 84. Le processeur lit le dernier caractère en même temps que le dernier caractère d'état, et il vérifie ce dernier afin de déterminer si un état de surcharge existe ou si un état d'abandon de colonne d'erreur de zone FCS a été détecté. Également, le caractère d'état se trouve vérifié afin de déterminer la zone d'un dernier multiplet partiel et de la longueur du dernier multiplet. Le processeur met alors fin à la séquence de colonne en lisant le caractère de contrôle de zone FCS (deux multiplets de 8 bits) à la sortie de la mémoire 84 (si cela est spécifié).

Explication des fig. 1 à 12

Les blocs logiques représentés aux fig. 1 à 12 ne seront pas décrits ici en détail élément par élément, afin d'éviter d'allonger et de compliquer inutilement la description de l'invention. Les blocs représentés sur ces figures se retrouvent dans le schéma de la fig. 14 et le tableau qui suit indique, pour chaque circuit représenté à la

fig. 14, les blocs correspondants montrés aux fig. 1 à 12. Sur ces dernières figures, les blocs sont désignés par des codes de référence tels que, par exemple, 01B01 et 08B04. Les deux derniers chiffres de ces codes indiquent la figure sur laquelle apparaît le bloc ainsi désigné. Ainsi, tous les blocs dont le code se termine par 01 apparaissent sur la fig. 1, tous les blocs dont le code de référence se termine par 04 apparaissent sur la fig. 4, etc.

Tableau de correspondance

Circuits de la fig. 14	Circuits correspondants des fig. 1 à 12
21	01B01, 01C01
22	06A06
24	01A06
26	01B06, 03B06
	05B06, 07B06
28	02C06, 06C06
	01A08 à 08A08
30	01A07, 04A07
	07A07
32	01C07, 04C07
	07C07, 10C07
	07D07, 10D07
34	01B08
36	07B04, 09A08
	10A08, 12A08
	07B08, 09B08
	03D08, 07C08
	05D08, 06D08
	07D08, 09C08
	09D08, 11C08
	12C08, 13C08
	11A06, 13A06
38	03C08
40	02C08, 01D08
	02D08
42	03A05, 03B05
44	04C05, 04D05
	05D05
46	04A06, 12A06
	10B06, 11C06
	12C06, 11D06
	12D06
48	01A02, 04A02
	07A02, 08A02
	10A02
	01B02 à 10B02
	01C02, 03C02
	07C02, 09C02
	01D02, 02D02
	03D02, 04D02
	09D02, 01B03
	01C03, 04B03
	04C03, 08B03
	08C03, 08D03
	10B03, 10C03
	10D03, 07A04
	08A04, 11A04
	08B04
50	01C05, 01D05
52	02C05, 02D05
54	03D05, 06C05
	06D05, 07D05
	08C05, 08D05
	09D05
56	08B01, 10C01
	10D01, 09A02

Circuits de la fig. 14	Circuits correspondants des fig. 1 à 12
	11B02, 12B02 02A03, 02B03 01D03, 02D03 04A03, 06A03 06B03, 04D03 05D03, 11C04 10A05, 11A05 12A05, 12X07 12X09
58	11B12, 13B12 11C12, 11D12
60	07D02, 08D02
62	01A10, 05A10
64	01B10 à 08B10
66	01C10, 07C10
68	01A09, 04A09 07A09
70	01C09, 04C09 07C09, 10C09 04D09, 07D09 10D09
72	06D10, 12D10 06A11, 08A11 09A11, 13A11 03B11, 06B11 07B11, 08B11 09B11, 03C11 02D11, 06C11 04D11, 08C11 08D11, 09D11 10D11, 12B12
74	12D04, 01C11 02C11
76	04A05, 04B05
78	05A05, 05B05
80	04C11
82	01B04, 04B04 01D04 à 08D04 02C12, 06C12
84	01B12, 03B12 06B12, 08B12
86	03A04, 05A04 06A04
88	04A12
90	01A05, 01B05 02A05, 06A05 06B05, 07A05 07B05, 08A05 08B05, 12C05
100	02D01

Les interconnexions entre les blocs représentés aux fig. 1 à 12, qui ne sont pas spécifiquement représentés par une ou des lignes de connexion, sont identifiées par un signal tel que, par exemple, le signal CNTRL1-00 indiqué à l'entrée 01 du connecteur 01B01 de la fig. 1a. La source de ce signal, ainsi que celle de tous les autres signaux, peut être aisément déterminée en consultant le tableau des fig. 20a à 20c. Ainsi, par exemple, en se reportant à la fig. 20a (colonne 1, ligne 14), on voit que le signal CNTRL1-00 a pour origine le bloc logique 01B01 de la fig. 1a.

Les blocs logiques des fig. 1 à 12 représentent des boîtiers de circuits intégrés normalisés connus, disponibles dans le commerce. Le code d'identification de chaque boîtier individuel est indiqué dans chaque bloc. Par exemple, la bascule JK 07D02 (fig. 2b) est un

boîtier de circuit intégré 74LS112 normalisé. Certains des circuits représentés sont disponibles uniquement chez des fournisseurs particuliers. Les microprocesseurs 01B08, 01C110 et 07C10 sont des composants similaires 2901 de Advanced Micro-Devices (A.M.D.). De même, les mémoires mortes programmables représentées aux fig. 7b et 9b sont des composants 82S137 de Signetics Corporation.

En consultant les fig. 1 à 12, il est recommandé que chacune des feuilles a et b composant chaque figure soit disposée côte à côte avec la partie a située à gauche de la partie b.

10 Description détaillée du mode de réalisation préféré

Le fonctionnement de l'adaptateur de ligne de communications selon l'invention va être décrit ci-après en se référant aux fig. 1 à 12 mais, avant d'entamer cette description générale, on donne ci-15 dessous une brève description des microprogrammes de la programmation câblée, utilisés dans le système.

Microprogramme câblé de transmission

Une ligne de programme comporte une première colonne com-20 mençant par un signe \$ qui indique le début d'une instruction, suivi d'un identificateur d'adresse. Cet identificateur est utilisé comme référence dans les instructions de branchement. Une deuxième colonne contient l'instruction elle-même. Il y a dix instructions de base dans le programme. Une troisième colonne est réservée aux25 zones opérande de l'instruction. Dans l'exemple décrit, il y a cinq opérandes commençant par 0 et se terminant par QR. Une quatrième colonne commençant par le signe (\$) est la zone commentaires de l'instruction, et des cinquième, sixième et septième colonnes concernent l'adresse hexadécimale de l'instruction dans la mémoire30 morte programmable PROM. Les sixième et septième colonnes concernent le contenu, en hexadécimal, à l'adresse indiquée. Le microprogramme de transmission a des instructions de 24 bits. Plus particulièrement, le contenu à l'adresse 070000 (hexadécimal) se compose comme suit: 0000 suivi de 0111 suivi de 0000 suivi de 0000 suivi de35 0000 suivi de 0000. Ce contenu correspond aux bits spécifiés comme bits 00 à 23, respectivement.

Une instruction MMP effectue un changement soit dans le contenu de la mémoire à accès sélectif du microprocesseur utilisée dans une opération de transmission, soit un changement dans le40 gistre Q du microprocesseur.

On notera que le microprocesseur utilisé pendant une opération de transmission est un microprocesseur à 4 bits, comme montré à la fig. 17a. Dans une opération de réception, deux microprocesseurs à 4 bits sont utilisés en cascade pour former un microprocesseur à 845 bits, comme montré à la fig. 17b. Une instruction TAMN a pour fonction de vérifier et modifier une micro-instruction suivante. En fonctionnement, cette instruction d'essai exécute un essai sur une fonction spécifiée par un opérande de zone test et, sur base du résultat de ce test, une micro-instruction suivante est soit exécutée, soit50 ignorée.

Une instruction SC se réfère à une sous-instruction. Cette instruction met à l'état un, à l'état initial ou charge, comme indiqué par sa zone opérande, des bascules indiquées des circuits.

Une instruction BR concerne une instruction de branchement55 pour passer d'une adresse en cours à une adresse indiquée, spécifiée dans une zone opérande BRADDR.

Une instruction BRSC est une instruction de branchement de sous-instruction qui combine les fonctions des instructions SC et BR.

Une instruction UDSCR se rapporte à la mise à jour d'un registre de commande d'ensemble de données. En fonctionnement, cette instruction commande l'écriture d'un signal du processeur de communications dans un registre de l'adaptateur de lignes de communi-60 cations.

Une instruction RCRCR vise la remise à l'état initial du registre de contrôle de parité cyclique. Cette instruction a pour effet de charger le registre de contrôle de parité cyclique des circuits de transmission avec des bits un.

Une instruction UCRRCR concerne une mise à jour du registre de contrôle de parité cyclique. En fonctionnement, le bit de données suivant est appliqué à ce registre.

Une instruction RER concerne une lecture de registre extérieur (registre extérieur du microprocesseur). L'information qui a ainsi été reçue du processeur de communications est introduite dans le microprocesseur.

Une instruction ZERO est simplement un moyen pour mettre à l'état zéro tous les emplacements inutilisés du programme câblé.

On décrit à présent les zones opérande de l'ensemble d'instructions. Dans chaque zone opérande, une zone WTFLD (zone type de mot) distingue une instruction de l'autre au bénéfice du matériel. Par exemple, en ce qui concerne le matériel, les instructions BR et BRSC sont les mêmes. Les instructions comprises dans le groupe commençant par l'instruction MMP et se terminant par l'instruction RER diffèrent entre elles en ce qui concerne la zone type de mot. Par exemple, l'instruction MMP a une zone opérande (WTFLD/0) indiquant que la zone type de mot contient uniquement des bits zéro. L'instruction TAMN indique que la zone type de mot à 3 bits contient un code 001. Pour l'instruction SC, la zone type de mot a un contenu 010. La zone type de mot de l'instruction RER a un contenu 111. La zone type de mot commence avec le bit 16 et a une longueur de 3 bits. Plus particulièrement, les bits 16, 17 et 18 de n'importe quel emplacement de la mémoire morte programmable constituent la zone type de mot. A la fig. 8, le décodeur 09B08 décode les bits 16, 17 et 18 pour indiquer un des huit types d'instructions, c'est-à-dire les instructions 0 à 7.

Chacune des instructions a deux lignes de zones opérande. La première ligne se rapporte aux zones qui sont spécifiées lorsque chaque instruction est écrite. La seconde ligne indique des zones opérande définies en permanence, qui ne sont pas spécifiées au moment où l'instruction est écrite. La zone type de mot apparaît dans la seconde ligne des zones opérande, et il s'agit donc d'une zone qui est établie en permanence et n'est pas modifiée par le programme. Une seconde zone opérande dans la seconde ligne de zones opérande est la zone ISCFLD qui apparaît dans le cas des instructions SC, BR et BRSC. La zone ISCFLD est une zone de 1 bit servant à inhiber une sous-instruction. Le contenu de cette zone est un zéro logique dans le cas d'une instruction SC ou BRSC qui autorise l'exécution de la sous-instruction. Dans l'instruction BR seulement, toutefois, la zone ISC contient un bit un qui inhibe la sous-instruction. Cette zone ISC est le bit 12 des emplacements de la mémoire morte programmable.

A la fig. 8, on voit que le bit de la zone ISC (bit TUCM13 + 00) est appliqué à la borne 4 du décodeur 09C08.

Une zone IFWFLD utilisée dans les instructions BR et BRSC est une zone inhibée écriture fichier. Dans chacune de ces instructions, la zone IFWFLD à 2 bits est à l'état logique zéro. Cette zone commence par le bit zéro. Les deux bits de cette zone sont appliqués au microprocesseur 01B08 (fig. 8). Plus particulièrement, les bits sont appliqués aux bornes 6 et 7 du microprocesseur. Un zéro logique apparaissant aux bornes 5, 6 et 7 du microprocesseur 01B08 empêche l'écriture d'informations dans la mémoire à accès sélectif du microprocesseur.

Une zone KFLD est une zone constante qui sert à la fois comme zone invariable dans la seconde ligne de zones opérande pour l'instruction RER, et comme zone variable dans maintes autres instructions telles que l'instruction MMP. La zone K constitue un moyen pour délivrer une valeur constante au système en vue de fonctions logiques ou arithmétiques. La zone K est une zone de 4 bits commençant par le bit 20.

Dans le cas de l'instruction RER, la zone K est une zone contenant uniquement les bits un.

Le microprogramme de transmissions réside dans les six mémoires mortes programmables 01C07, 04C07, 07C07, 10C07, 07D07, 10D07 (fig. 7). Les vingt-quatre sorties de ces mémoires sont notées TUCM00 à TUCM23.

On va décrire les zones opérande qui sont utilisées lorsque chaque instruction est exécutée.

La première zone est la zone KFLD, appelée zone constante. Cette zone est utilisée pour permettre au programme d'introduire une constante ou un masque d'édition dans l'opération. La zone KFLD occupe les bits 20 à 23. Cette zone se trouve réfléchie dans le circuit de la fig. 8 par des signaux TUCM20, TUCM21, TUCM22 et TUCM23. Ces signaux se reproduisent à la sortie de la mémoire morte programmable 07D07. Les signaux de la zone constante sont appliqués aux entrées données du microprocesseur 01B08 par l'intermédiaire des nœuds 07A08, 05A08, 03A08 et 01A08 (fig. 8) qui réalisent des fonctions ET.

La zone KFLD est utilisée dans les instructions suivantes: MMP, TAMN, SC, UDSCR, RCRRCR, UCRRCR.

Des zones FAFLD et FBFLD sont utilisées pour adresser la mémoire à accès sélectif interne de 64 bits du microprocesseur 01B08. Cette mémoire comprend seize mots de 4 bits. Ses deux sorties sont identifiées par les lettres A et B à la fig. 17a. Afin d'adresser la mémoire à accès sélectif interne, deux zones adresse sont par conséquent requises, à savoir la zone FA et la zone FB. Ces zones sont appliquées respectivement aux entrées adresse A et adresse B de la mémoire à accès sélectif et elles commandent les sorties A et B. Les zones FAFLD et FBFLD sont des zones de 4 bits chacune. La zone FAFLD occupe les bits 12 à 15, tandis que la zone FBFLD occupe les bits 8 à 11. Se reportant aux quatre colonnes associées à chacune de ces zones, par exemple, lorsqu'un identificateur de programme A0 est utilisé par un programmeur, l'assembleur de programme engendre un zéro hexadécimal. D'une manière similaire, si un identificateur de programme A4 est utilisé, l'assembleur de programme engendre un 4 hexadécimal (0100). Ces nombres hexadécimaux apparaissent dans les bits 12 à 15. Les signaux de la zone A apparaissent à la sortie de la mémoire morte programmable 1C07, tandis que les signaux de la zone B apparaissent à la sortie de la mémoire morte programmable 04C07. Les signaux de ces zones sont appliqués respectivement aux entrées adresse ARAM et BRAM du microprocesseur 01B08.

La zone FAFLD est utilisée avec les instructions suivantes: MMP, PAMN, UDSCR, RCRRCR, UCRRCR. La zone FBFLD est utilisée avec les instructions suivantes: MMP, USC, UDSCR, RCRRCR, UCRRCR et RAR.

Une zone OPFLD est la zone code opération. Cette zone occupe deux emplacements différents: les bits 2 à 7 et le bit 19. Les bits de cette zone spécifient la fonction arithmétique ou logique à exécuter par le microprocesseur. En ce qui concerne les fonctions logiques, par exemple, une fonction AORB indique que les données apparaissant dans la mémoire à accès sélectif d'entrée du microprocesseur adressée par les entrées adresse sont combinées logiquement par fonction OU avec les données apparaissant à l'emplacement adressé par les entrées adresse B du microprocesseur. La fonction indique, entre autres, par un C dans la seconde colonne qu'une valeur à traiter est une valeur octale. La valeur octale est appliquée aux bits 2 à 7 et au bit 19. Les fonctions logiques qui doivent être exécutées sont décrites dans une zone commentaire de la zone code opération, OPFLD. Ces commentaires se présentent également dans la troisième colonne des fonctions arithmétiques.

Par exemple, la lettre D indique les données sur les lignes d'entrée de données du microprocesseur 01B08, et la lettre Q désigne les données stockées dans le registre Q interne du microprocesseur. Le terme AND désigne une fonction logique d'interception (fonction ET), le terme NOT désigne une inversion logique (fonction NON), le terme XOR désigne une fonction logique OU-exclusif, le terme XNOR désigne une fonction logique NI et le terme ALUZ indique que l'unité arithmétique et logique interne du microprocesseur est dans l'état où sa sortie contient uniquement des zéros.

En ce qui concerne les fonctions arithmétiques de la zone code opération, la lettre P indique une opération d'addition ou opération a +, la lettre M indique une opération de soustraction ou opération a -, le code 1C indique un complément à 1, et 2C indique un com-

plément à 2. Par exemple, une fonction QM1 indique qu'un 1 est soustrait du contenu du registre Q du microprocesseur.

Les signaux de la zone code opération apparaissent aux sorties des mémoires mortes programmables 10C07, 07C07 et 10D07. Ces signaux sont les signaux TUCM02 à TUCM07 et TUCM19. Ces signaux sont appliqués aux entrées bits d'instruction 0 à 5 du microprocesseur 01B08 et à l'entrée report CN du microprocesseur.

La zone OPFLD est utilisée pour l'exécution de toutes les instructions sauf les instructions BR, BRSC et O.

Une zone FDFLD est la zone fichier destination. Cette zone contient 2 bits et commence au bit zéro. Cette zone désigne l'endroit où la sortie de l'unité arithmétique et logique du microprocesseur est stockée et si un décalage doit être produit avant de stocker la donnée. Par exemple, une instruction QR indique que la donnée est stockée dans le registre Q du microprocesseur sans décalage. Une instruction FR indique que la sortie de l'unité arithmétique et logique est stockée dans la mémoire à accès sélectif interne à l'adresse désignée par les lignes d'entrées adresse BRAM du microprocesseur. Une instruction SRQF indique que le contenu du registre Q doit être décalé vers la droite, puis stocké de nouveau dans ce même registre, et que la sortie de l'unité arithmétique et logique est chargée dans la mémoire à accès sélectif interne après avoir été décalée vers la droite. Une instruction SLQF indique que l'information stockée dans le registre Q est décalée vers la gauche de 1 bit et ensuite stockée dans le registre Q, et que la sortie de l'unité arithmétique et logique est décalée de 1 bit vers la gauche, puis stockée dans la mémoire à accès sélectif interne.

A la fig. 8, le signal TQRQZZ à la sortie S1 du microprocesseur 01B08 indique le bit d'extrême droite du registre Q. La sortie S1 est connectée à l'entrée S4 ou entrée d'extrême gauche de la mémoire à accès sélectif interne du microprocesseur (fig. 17a). Ainsi, pendant l'exécution d'une instruction SRQF, la sortie S1 est appliquée à l'entrée S4 du microprocesseur et, pendant l'exécution d'une instruction SLQF, la sortie S4 est appliquée à l'entrée S1 du microprocesseur.

On notera que les broches S1 et S4 sont bidirectionnelles. De cette manière, un registre à décalage de 8 bits a été formé en connectant le registre Q de 4 bits avec un emplacement de 4 bits dans la mémoire à accès sélectif interne. Pendant l'exécution d'une instruction SRQF, la sortie de la porte — ET 03D08 est appliquée à l'entrée S2 du microprocesseur. Un caractère de contrôle de parité cyclique engendré par le générateur 03C08 est ainsi introduit par décalage dans le microprocesseur. Les 2 bits constituant la zone FDFLD correspondent aux signaux TUCM00 et TUCM01 aux sorties de la mémoire morte programmable 10C07. Ces signaux sont appliqués aux bits d'instruction 7 et 8 du microprocesseur 01B08. Cette zone FDFLD est utilisée par chacune des instructions, sauf les instructions BR, BRSC et O.

Une zone TSTFLD est une zone test utilisée uniquement par l'instruction TAMN. La première colonne identifie sept fonctions câblées qui peuvent être testées par l'instruction TAMN. La zone test comprend 4 bits: les bits 8 à 11. Les 4 bits correspondent aux signaux TUCM08 à TUCM11 à la sortie de la mémoire morte programmable 04C07. Les signaux TUCM09 à 11 sont appliqués aux entrées sélection du multiplexeur 10A08 (fig. 8). Lorsque les signaux TUCM08 à TUCM11 sont à l'état zéro, cela indique que le bit de transmission suivant est prêt à être testé (fonction TRDY). S'il est constaté que le bit n'est pas prêt, l'instruction suivante est exécutée. Par contre, si le bit suivant est prêt à être transmis, l'instruction suivante est traitée comme une instruction pas d'opération. Une fonction TFMNE est vérifiée lorsqu'un 3 octal apparaît sur les lignes TUCM08 à 11. Plus particulièrement, si la mémoire FIFO de transmission est vide, c'est l'instruction suivante qui est exécutée. Si cette mémoire n'est pas vide, l'instruction suivante est traitée comme une instruction pas d'opération. La zone TSFLD est utilisée uniquement avec l'instruction TAMN.

Une zone SCFLD est la zone sous-instruction utilisée uniquement par une instruction SC. Cette zone comprend 3 bits commen-

çant au bit 13, et elle spécifie l'une de huit fonctions différentes. Les signaux TUCM13 à TUCM15 correspondant à cette zone apparaissent aux sorties des mémoires mortes programmables 01C07 (fig. 7) et y sont appliqués aux entrées sélection du décodeur 09C08. L'apparition d'un zéro hexadécimal sur les lignes TUCM13-15 indique l'occurrence d'une fonction de remise à zéro du bit transmission prêt. L'apparition d'un un hexadécimal sur les lignes TUCM13-15 indique un signal écriture tampon bit de transmission, comme indiqué par la zone commentaire à la colonne 3.

On voit ainsi que les signaux TUCM13-15 appliqués au décodeur 09C08 portent une des sorties du décodeur à l'état logique zéro. Cela se produit pendant l'exécution d'une instruction SC ou d'une instruction BRSC. La zone SCFLD est utilisée pendant les instructions SC et BRSC.

Une zone XRFLD est la zone registre extérieur utilisée uniquement pendant l'exécution de l'instruction RER. Cette zone spécifie un parmi cinq registres différents devant être copiés dans le microprocesseur 01B08, par l'intermédiaire des lignes entrée données du microprocesseur. La zone XRFLD est spécifiée par les bits 12 à 15 indiqués par les lignes TUCM12 à TUCM15 à la sortie de la mémoire morte programmable 01C07. Le signal TUCM12 est appliqué aux entrées sélectrices des multiplexeurs 02C06 et 06C06 (fig. 6). Les signaux TUCM13-15 sont appliqués respectivement aux portes ET 05D08, 06D08 et 07D08.

Une zone BRADDR, utilisée uniquement par les instructions BR et BRSC, désigne un emplacement auquel un branchement doit être effectué. L'adresse de branchement contenue dans cette zone comporte 10 bits commençant par la position binaire 2. Se reportant à la fig. 7, on voit que la zone BRADDR est caractérisée par les signaux TUCM02 à TUCM11 aux sorties des mémoires mortes programmables 10C07, 07C07 et 04C07. Ces signaux, à leur tour, sont appliqués aux compteurs à 4 bits 01A07, 04A07 et 07A07, afin d'adresser les six mémoires portes programmables de la mémoire de commande de programme de la fig. 7.

35 Programmation câblée de réception

Le format du microprogramme de réception est semblable au microprogramme de transmission et seules les différences entre les programmes, qui peuvent être rencontrés par l'homme de l'art, vont être soulignées ici. Le microprogramme de réception est stocké dans les mémoires mortes programmables illustrées à la fig. 9, plus particulièrement les mémoires mortes programmables 01C09, 04C09, 07C09, 10C09, 04D09, 07D09 et 10D09. Il est à noter que, dans le cas du microprogramme de réception, il y a sept mémoires mortes programmables au lieu de six pour le microprogramme de transmission. La mémoire morte programmable additionnelle est nécessaire car le mot de commande réception est un mot de 28 bits au lieu de 24 bits pour le mot de commande de transmission. Le récepteur utilise deux microprocesseurs de 4 bits en cascade (fig. 17b) pour former un trajet de données à 8 bits dans le récepteur. La mémoire morte programmable additionnelle utilisée par le récepteur correspond aux 4 bits additionnels requis par la zone D utilisée par le récepteur. Les deux microprocesseurs utilisés par le récepteur sont des microprocesseurs 01C10 et 07C10 de la fig. 10.

Les bits de chaque zone opérande correspondent aux signaux RUCM du récepteur. Par exemple, une zone KFLD indique la zone K a une longueur de 8 bits commençant par le bit 20 dans la mémoire de commande de programme réception. Ces bits correspondent aux signaux RUCM20 à RUCM27 aux sorties des mémoires mortes programmables 07D09 et 04D09 (fig. 9).

Les signaux RUCM sont appliqués par les nœuds 01B10 à 08B10 (fig. 10) aux entrées de données des microprocesseurs 01C10 et 07C10. Plus particulièrement, les signaux RUCM20 à RUCM23 sont appliqués aux entrées du microprocesseur 01C10, tandis que les signaux RUCM24 à RUCM27 sont appliqués aux entrées du microprocesseur 07C10. Des signaux FAFLD et FBFLD sont appliqués aux microprocesseurs 01C10 et 07C10. Plus particulièrement, les signaux FAFLD sont appliqués aux entrées adresse de mémoire à

accès sélectif A des deux microprocesseurs, tandis que les signaux FBFLD sont appliqués aux entrées adresse de mémoire à accès sélectif B des microprocesseurs. Les signaux d'une zone OPFLD et les signaux FDFLD sont appliqués aux microprocesseurs, les signaux OPFLD correspondant aux signaux RUCM02 à RUCM07 qui sont appliqués aux bits d'instructions 0 à 5 du microprocesseur 01C10 et aux bits d'instructions 0 à 5 du microprocesseur 07C10. Le bit 19 de la zone OPFLD correspond au signal RUCM19 qui se trouve appliqué à l'entrée report CN du microprocesseur 07C10 seulement. L'entrée report du microprocesseur 01C10 est connectée à la sortie report du microprocesseur 07C10. Les bits 0 et 1 de la zone F sont appliqués aux bits d'instructions 7 et 8 de chacun des microprocesseurs.

Line	ARGDEF	SCFLD	(13/3)
16300	RSRRDY	/0	RESET RCX BIT READY FF
16400	RSRCRSYN	/1	RESET RECEIVE RESYNC FF
16500	WRRFMS	/2	WRITE RCX FIFO MEMORY WITH STATUS
16600	WRRFMD	/3	WRITE RCX FIFO MEMORY WITH DATA
16700	RSRFMB	/4	RESET RCX FIFO MEMORY BUFFER
16800	WRRFMBS	/5	WRITE RCX FIFO MEMORY BUFFER WITH STATUS
16900	WRRFMBD	/6	WRITE RCX FIFO MEMORY BUFFER WITH DATA
17000	PSRFCSR	/7	PRESET RCX FCS REGISTER

Une autre différence entre les microprogrammes de réception et de transmission est indiquée dans la zone FDFLD du microprogramme de réception. Les instructions SRQF et SLQF sont appliquées différemment dans les microprocesseurs de réception 01C10 et 07C10. Par exemple, pendant l'exécution de l'instruction SRQF, le bit d'extrême droite du registre Q du microprocesseur est appliqué à l'entrée S2 connectée au bit d'extrême gauche du registre Q du microprocesseur 07C10. En outre, le bit d'extrême droite de la mémoire à accès sélectif interne du microprocesseur 01C10 est appliqué par l'accès S3 du microprocesseur à l'entrée S4 du microprocesseur 07C10, qui se trouve connectée au bit d'extrême gauche de la mémoire à accès sélectif interne de ce microprocesseur.

En outre, le bit d'extrême droite du registre Q du microprocesseur 07C10 est appliqué par l'accès S1 à l'entrée S4 du microprocesseur 01C10, entrée qui se trouve connectée au bit d'extrême gauche de la mémoire à accès sélectif interne du microprocesseur 01C10. Un registre à décalage de 16 bits est ainsi formé, qui comprend un registre Q de 8 bits et une mémoire à accès sélectif de 8 bits qui travaille tant pour les opérations de décalage à gauche que pour les opérations de décalage à droite. Pendant une opération de décalage à droite (opération SRQF), un bit de données de réception est reçu de l'équipement de communications de données à l'entrée S2 du microprocesseur 01C10, et ce bit est introduit dans la position binaire d'extrême gauche du registre Q de ce microprocesseur.

Pendant une opération de décalage à gauche (opération SLQF), la sortie de la porte NON-ET 04D11 (fig. 11) est appliquée par le nœud 12D10 à l'entrée S3 du microprocesseur 07C10, qui se trouve connectée à la condition binaire d'extrême droite de la mémoire à accès sélectif interne de ce microprocesseur. On voit donc ainsi que, pendant une opération de décalage à droite, les signaux apparaissant aux accès S1 et S3 du microprocesseur 01C10 sont appliqués aux accès S2 et S4, respectivement, du microprocesseur 07C10. Pendant une opération de décalage à gauche, les signaux apparaissant aux accès S2 et S4 du microprocesseur 07C10 sont appliqués respectivement aux accès S1 et S3 du microprocesseur 01C10. L'accès S4 du microprocesseur 01C10 et l'accès S1 du microprocesseur 07C10 sont également bidirectionnels.

Fonctionnement du système Phase d'initialisation

L'initialisation de l'adaptateur de ligne commence lorsqu'un signal maître à l'état initial est reçu du processeur de communica-

Des zones TSTFLD et SCFLD seront décrites ultérieurement au cours de la description du fonctionnement du système. Les bits 8 à 11 de la zone TSTFLD correspondent aux signaux RUCM08 à RUCM11 appliqués aux entrées adresse de mémoire à accès sélectif B des microprocesseurs 01C10 et 07C10; les bits 13 à 15 de la zone SCFLD correspondent aux signaux RUCM13 à 15 appliqués aux entrées adresse de mémoire à accès sélectif A des microprocesseurs. Les signaux RUCM13 à 15 sont appliqués aux entrées d'un décodeur 08C11 (fig. 11). Les fonctions câblées spécifiées par la zone TSTFLD sont décrites dans la zone commentaire. La zone SCFLD peut spécifier un parmi huit nombres. L'effet de l'application d'un tel nombre aux lignes RUCM13 à 15 ressort de la liste de programme suivante, où le code RCX désigne un récepteur.

tions à la broche 24 du connecteur 01B01 (fig. 1). Ce signal est ensuite appliqué par l'inverseur 10A02 (fig. 2) et par les portes — NI 09B02 et 10B02 aux registres et bascules de remise à l'état initial de l'adaptateur. En outre, les compteurs 01A07, 04A07 et 07A07 (fig. 7) sont remis à l'état zéro. Les compteurs comprennent un compteur de 10 bits fournissant les adresses à la mémoire de commande de transmission comprenant les mémoires mortes programmables 01C07, 04C07, 07C07, 10C07, 07D07 et 10D07. Le microprogramme de transmission est ainsi initialisé et passe à l'exécution.

Un signal d'horloge de 238 ns est reçu du processeur de communications à la broche 18 du connecteur 01B01 et est appliqué aux entrées de la porte — OU 04D03 (fig. 3). La sortie de cette porte est transmise à travers l'inverseur 04C03 et distribuée à travers tout le système. En particulier, le signal d'horloge est appliqué aux entrées d'horloge des compteurs 01A07, 04A07 et 07A07. En réponse à ce signal d'horloge, les compteurs sont normalement incrémentés pour adresser des emplacements séquentiels dans la mémoire de commande de programme de transmission.

Les bits TUCM16, 17 et 18 à la sortie de la mémoire morte programmable 10D07 désignent lequel des huit types d'instructions doit être exécuté. Les bits sont appliqués au décodeur 09B08 (fig. 8) afin de porter à l'état logique zéro une des huit lignes de sortie de ce décodeur. Pendant une instruction de branchement (BR), ou une instruction de branchement de sous-programme (BRSC), la ligne de sortie du bit 3 du décodeur 09B08 passe à l'état logique zéro qui se trouve appliqué aux entrées de chargement des compteurs de programme de transmission de la fig. 7. L'adresse de branchement à 10 bits désignée par les signaux TUCM02 à TUCM11 est alors chargée dans les compteurs de programme de transmission. Le microprocesseur passe de l'emplacement 010 hexadécimal à l'emplacement 022 hexadécimal plutôt qu'à l'instruction séquentielle suivante indiquée par le code 011 hexadécimal.

Il est entendu que le microprogramme de transmission reste en une seule boucle de programme, jusqu'à ce que le processeur de communications fournisse l'information d'initialisation en plus du signal maître à l'état initial.

Le signal maître à l'état initial reçu du processeur de communications est également appliqué aux entrées de remise à zéro des compteurs 01A09, 04A09 et 07A09 (fig. 9). Ces compteurs constituent les compteurs de programme de réception destinés à adresser la mémoire de commande de programme de réception constituée des mémoires mortes programmables 01C09, 04C09, 07C09, 10C09,

04D09, 07D09 et 10D09. Les compteurs de programme sont alors mis dans l'état par lequel ils indiquent une ligne spécifiée du micro-programme de réception. Les bits RUCM16, 17 et 18 à la sortie de la mémoire morte programmable 10D09 désignent le type d'instructions qui doit être exécuté. Ces bits sont appliqués au décodeur 06A11 (fig. 11) pour porter à l'état logique zéro une des huit sorties du décodeur. Pendant une instruction BR ou BRSC, la sortie de bit 3 du décodeur est portée à l'état logique zéro. Les compteurs 01A09, 04A09 et 07A09 sont ensuite chargés avec les bits indiqués par les signaux RUCM02 à RUCM11 indiquant une adresse suivante dans la mémoire de commande de programme de réception. Une adresse autre que l'adresse séquentielle suivante est ainsi fournie pendant une instruction de type de branchement.

Dans les trois dernières colonnes, d'un exemple d'instruction de branchement, un branchement est effectué de l'emplacement 009 hexadécimal vers l'emplacement 012 hexadécimal plutôt que vers l'emplacement séquentiel suivant 00A. Après réception du signal maître à l'état initial, le récepteur reste dans une boucle de programme, jusqu'à ce qu'une autre information d'initialisation soit reçue. Le processeur de communications détermine ensuite le type d'adaptateur de lignes auquel il est connecté. Comme le processeur de communications est connecté à plusieurs unités autres que l'adaptateur de lignes BHCLA, le processeur de communications doit fournir une adresse particulière à l'adaptateur de lignes. Par exemple, les signaux d'adresse ADDRS1 à ADDRS3 se trouvent appliqués par le processeur de communications aux broches 4 à 6 du connecteur 01B01. Le signal ADDRS3 doit être à l'état zéro pour adresser soit le décodeur d'adresse de transmission, soit le décodeur d'adresse de réception. Le décodeur d'adresse de transmission est le décodeur 01A02 (fig. 2), et le décodeur d'adresse de réception est le décodeur 04A02. En outre, le signal ADDRS2 doit être à l'état logique un et le signal ADDRS1 doit être à l'état logique zéro pour adresser le décodeur d'adresse transmission. Si le signal ADDRS2 est à l'état logique un et le signal ADDRS1 + 10 à l'état logique zéro, c'est le décodeur d'adresse réception qui est adressé.

Le processeur de communications fournit ensuite les signaux de commande indiqués par les signaux CNTRL1 à CNTRL3, respectivement appliqués aux broches 1 à 3 du connecteur 01B01, afin d'indiquer le type de fonction que doit exécuter l'adaptateur de lignes BHCLA. Fondamentalement, ces signaux de commande indiquent quels registres doivent être chargés ou vidés. Les signaux de commande sont appliqués aux décodeurs 01A02 et 04A02. Pendant une phase d'opération du processeur de communications, par conséquent, une entrée d'un des deux décodeurs passe à l'état logique zéro. Comme plusieurs dispositifs sont connectés au processeur de communications, celui-ci doit d'abord déterminer quel type de dispositif est connecté à cette adresse. Un état logique un sur chacune des lignes de commande fait passer à l'état logique zéro la sortie binaire 7 du décodeur. Cela peut se produire soit au décodeur de transmission, soit au décodeur de réception. Lorsque l'état logique zéro apparaît à la sortie binaire 7 de l'un ou l'autre des deux décodeurs, cet état se trouve appliqué à la porte —OU 01D02 dont la sortie passe à l'état logique zéro. La sortie de la porte —OU 02D02 passe alors à l'état logique zéro, et la sortie de la porte 01D02 est appliquée aux entrées sélection des multiplexeurs 01B04 et 04B04. Lorsqu'un état logique zéro est appliqué à l'entrée sélection, les multiplexeurs sont sélectionnés pour travailler sur leurs entrées 0. Les entrées 0 du multiplexeur 01B04 sont à l'état logique un. L'entrée à 4 bits du multiplexeur 04B04 est cependant à l'état logique 0110. Pour d'autres dispositifs, différents codes sont câblés dans les multiplexeurs afin d'identifier ces dispositifs.

La sortie de la porte 02D02 est appliquée aux entrées F (sortie validée) des multiplexeurs 01B04 et 04B04. Lorsque cette entrée est à l'état logique zéro, les codes décrits plus haut sont appliqués par les multiplexeurs aux lignes LADAT0 à LADAT7 et par les nœuds 01D04 à 08D04 (fig. 4b) au connecteur 01C01 de la fig. 1. Les signaux sont ainsi envoyés au processeur de communications pour identifier le dispositif adressé. Le processeur de communications dé-

termine alors le type de dispositif adressé et il peut ensuite procéder à la phase d'initialisation.

Le processeur de communications charge alors le registre de commande DCE constitué des registres à 4 bits 08A04 et 11A04. Les lignes d'adresse et de commande des décodeurs de la fig. 2 sont ainsi excitées, comme décrit plus haut. Selon celui des décodeurs qui se trouve validé, la sortie binaire 5 du décodeur passe à l'état logique zéro. La sortie de la porte —NI 03B02 passe dès lors à l'état logique un qui se trouve appliqué à la porte —ET 03C02. Le processeur de communications fournit ensuite une impulsion de pointage à la broche 25 du connecteur 01B01, qui se trouve envoyée à travers l'inverseur 08A02 à une seconde entrée de la porte —ET 03C02. La sortie de celle-ci passe alors à l'état logique zéro qui se trouve appliqué à l'entrée d'horloge des registres 08A04 et 11A04. Le processeur de communications applique des données de commande DCE aux broches 17, 18 et 20 à 25 du connecteur 01C01 et, de là, aux entrées données des registres 08A04 et 11A04. Lorsque ces registres sont synchronisés, les données sont chargées dans ces registres.

Périodiquement, le microprogramme commande au registre 08B04 d'enregistrer le contenu du registre 08A04. Cette instruction UDSCR fait passer la sortie binaire 4 du décodeur 09B08 (fig. 8) à l'état logique zéro. Cet état logique zéro se trouve appliqué à une entrée de la porte —ET 07B04. La sortie de celle-ci est appliquée à l'entrée d'horloge du registre 08B04 dont la sortie produit les signaux de commande à l'équipement de communications de données local. Ces signaux sont appliqués aux dispositifs d'excitation 01C05, 02C05, 06C05 et 08C05 (fig. 5), dont les sorties sont appliquées à l'équipement de communications de données local par l'intermédiaire des éléments de logique de la fig. 5 et du connecteur 02D01.

Pour achever l'opération d'initialisation, l'information d'état (fig. 15g) provenant de l'équipement de communications de données local est lue par le processeur de communications à travers l'adaptateur de lignes. Sous la commande du processeur de communications, la sortie binaire 2 du décodeur 01A02 ou du décodeur 04A02 passe à l'état logique zéro. Ces signaux sont appliqués à la porte —OU 03D02 dont la sortie passe à l'état logique zéro, de manière à porter à l'état logique zéro la sortie de la porte 02D02. Pendant ce temps, les entrées sélection des multiplexeurs 01B04 et 04B04 sont à l'état logique un afin de porter leurs entrées à l'état un. L'information d'état de l'équipement de communications de données local est appliquée au connecteur 02D01 et, de là, aux circuits 03A05 à 08A05 (fig. 5).

L'information d'état de l'équipement de communications de données local comprend les signaux DCERIZ, DCEDSR, CDECTS et DCECDZ qui sont appliqués par le connecteur 02D01 au récepteur 01A05, à une résistance 02A05 et aux récepteurs d'interface 06A05 à 08A05, respectivement. La sortie du récepteur 01A05 est appliquée par l'inverseur 01B05 à une entrée du multiplexeur 01B04. Les sorties des récepteurs 06A05 à 08A05 sont appliquées aux trois autres entrées du multiplexeur 01B04. Lorsque l'entrée F du multiplexeur 01B04 passe à un état logique zéro, les entrées niveau un du multiplexeur sont appliquées aux lignes de données de l'adaptateur de lignes 0 à 3 et envoyées au processeur de communications comme décrit plus haut. Celui-ci examine alors l'information d'état reçue afin de déterminer l'état fonctionnel de l'équipement de communications de données local. Selon l'état de fonctionnement de cet équipement de communications de données local, le processeur de communications lance une opération de transmission ou une opération de réception.

Opération de transmission

Une opération de transmission est lancée lorsque la sortie de la porte —ET 12A06 (fig. 6) passe à l'état logique un. Dans ce cas, les registres d'entrée de chacune des mémoires premier-entré premier-sorti 01B06, 03B06, 05B06 et 07B06 sont vides. La sortie de la porte —ET 12A06 est appliquée à une entrée de la porte —ET 11C06, et la sortie de celle-ci passe alors à l'état logique un. On notera que le

bit 1 du registre de commande DCE 08A04, qui stocke le bit RTS pour le mot de commande DCE (fig. 15f), doit être à l'état logique un et que le signal prêt à transmettre (fig. 15g) provenant de l'équipement de communications de données DCE (à la broche 25 du connecteur 02D01), qui est également appliqué au bit un du multiplexeur 01B04, doit être à l'état logique un appliqué à la porte 11C06 afin de valider sa sortie pour qu'elle passe à l'état logique un. La bascule JK 11D06 se trouve ainsi mise à l'état un et la sortie de la porte —ET12D06 passe à l'état logique zéro. La sortie de la porte 12D06 est appliquée à la broche 11 du connecteur 01B01, ce qui a pour effet d'indiquer au processeur de communications le début d'une opération de transmission.

En réponse à cette indication, le processeur de communications lance les signaux d'adresse et de commande, décrits plus haut, au décodeur d'adresse de transmission 01A02. La sortie bit 2 du décodeur passe alors à l'état logique zéro. Il est entendu que, pendant ce temps, la sortie bit 2 du décodeur d'adresse de réception 04A02 reste à l'état logique un, ainsi que toutes les autres sorties des décodeurs d'adresses 01A02 et 04A02.

La sortie de la porte 11C06 est également appliquée à une entrée de la porte 03A04. Comme indiqué plus haut, la sortie bit 2 du décodeur 04A02 est à l'état logique un, et celui-ci se trouve appliqué à une seconde entrée de la porte 03A04 dont la sortie est à l'état logique zéro, cet état se trouvant appliqué à travers l'inverseur A0504 à une entrée d'état un du multiplexeur 04B04. Comme un dispositif d'identification n'est pas lu par le processeur de communications à ce moment, l'entrée sélection du multiplexeur 04B04 se trouve à l'état logique un afin de sélectionner les entrées d'état un du multiplexeur. La sortie étage un du multiplexeur 04B04 se trouve donc à l'état logique un, qui se trouve appliqué par le nœud 05D04 à la broche 5 du connecteur 01C01. Le processeur de communications est ainsi informé que les registres d'entrée des mémoires de la fig. 6 sont prêts à recevoir les informations de commande et de transmission à transmettre.

Le processeur de communications charge alors le mot de commande étendu de transmission (fig. 15a) dans le registre (fig. 6). Plus particulièrement, le processeur de communications applique cette information, par l'intermédiaire des broches 22 à 25 du connecteur 01C01, aux entrées du registre 06A06. En outre, le processeur applique les informations d'adresse et de commande au décodeur 01A02 dont la sortie bit 1 se trouve portée à l'état logique zéro, tandis que les entrées du registre 06A06 se trouvent validées. Le processeur applique également un signal de pointage à l'entrée d'horloge du registre 06A06, par l'intermédiaire de la broche 25 du connecteur 01B01, afin de charger les données de commande dans le registre. On remarquera que l'information enregistrée dans le registre 06A06 est utilisée pendant l'opération de transmission, comme décrit plus loin.

Le processeur de communications applique ensuite des informations d'adresse et de commande au décodeur 01A02, afin de faire passer la sortie bit 0 de celui-ci à l'état logique zéro. Cet état est transféré par l'intermédiaire de la porte —ET 06B02 par un signal de pointage provenant du processeur de communications. Au moment où apparaît ce signal de pointage, le processeur de communications des informations de commande de transmission additionnelles sur les lignes de données aboutissant aux entrées du registre 01A06. La sortie zéro de la porte 06B02 se trouve appliquée à l'entrée d'horloge du registre 01A06 afin de charger dans ce registre le mot de commande de transmission (fig. 15b). Les sorties du registre 01A06 sont appliquées aux entrées des mémoires 01B06 et 03B06. Les mémoires 01B06, 03B06, 05B06 et 07B06 forment une mémoire tampon de transmission premier-entré premier-sorti. Cette mémoire a une largeur de 16 bits et une profondeur de 64 bits. Les mémoires 01B06 et 03B06 contiennent l'information de commande de transmission, tandis que les mémoires 05B06 et 07B06 contiennent les données à transmettre. Ainsi, pour chaque caractère de données fourni par le processeur de communications, celui-ci fournit également un caractère de contrôle.

Dans le cas où la sortie bit 3 du registre 01A06 se trouve à un état logique un, la sortie de ce registre est appliquée à un inverseur 04A06 afin de porter la sortie de la porte 11C06 à l'état logique zéro. Se trouve ainsi inhibée la transmission de tout autre signal d'interruption au processeur de communications.

A partir de ce moment, le chargement du registre 01A06 tel que décrit plus haut sera appelé chargement du registre de mot de commande de transmission. L'opération pendant laquelle le processeur de communications charge le registre 06A06 est appelée chargement du registre de mot de commande étendu. L'opération relative à la lecture de l'information d'état de l'équipement de communications de données DCE (fig. 15g) est appelée opération de lecture d'état.

Plus particulièrement, l'information de lecture d'état se produit lorsque le processeur de communications provoque l'apparition d'un état logique un à l'entrée sélection des multiplexeurs 01B04 et 04B04, et l'apparition d'un état logique zéro à l'entrée validation de sortie des multiplexeurs. A ce moment, les quatre signaux d'état DCE appliqués aux entrées d'état un du multiplexeur 01B04 sont envoyés par le multiplexeur au processeur de communications, par l'intermédiaire des nœuds 01D04 à 04D04. En même temps, les signaux aux entrées d'état un du multiplexeur 04B04 sont envoyés au processeur de communications, par l'intermédiaire des nœuds 05D04 à 08D04. La sortie de l'inverseur 05A04 est utilisée pendant les opérations de transmission et de réception, et sera décrite plus loin. Les deux autres entrées d'état un du multiplexeur 04B04 sont toujours à l'état logique zéro. La dernière entrée d'état un du multiplexeur 04B04 est connectée à la sortie Q de la bascule JK 06A04 et est utilisée pendant les opérations de transmission, comme on le verra plus loin.

On décrit à présent le chargement d'un premier caractère de donnée provenant du processeur de communications. Celui-ci applique la donnée aux lignes de données qui aboutissent aux entrées des mémoires 05B06 et 07B06. En même temps, le processeur applique les signaux d'adresse et de commande au décodeur 01A02, afin de porter la sortie bit 6 du décodeur à l'état logique zéro qui se trouve appliqué à l'inverseur 01C06 par l'intermédiaire de la porte NON-ET 01B02, la sortie de l'inverseur passant ainsi à l'état logique zéro.

La sortie de la porte 01B02 est également appliquée aux entrées chargement de la mémoire tampon de transmission de la fig. 6, chargeant ainsi la donnée transmise dans les mémoires 05B06 et 07B06. En outre, l'information de commande chargée précédemment par le processeur dans le registre 01A06 se trouve chargée dans les mémoires 01B06 et 03B06. De cette manière, un mot de commande de transmission se trouve chargé dans la mémoire tampon de transmission chaque fois qu'un mot de données se trouve chargé dans la mémoire tampon. Comme le mot de commande de transmission se trouve également stocké dans le registre 01A06, le registre ne doit pas être rafraîchi avant que le processeur de communications ne requière un mot de commande différent pour un mot de donnée différent. Ainsi, plus d'un mot de donnée peut être utilisé conjointement avec le même de commande. En particulier, le mot de commande ne sera pas changé avant qu'un dernier caractère du message transmis n'ait été reçu. Dans ce cas, la sortie bit 3 du registre 01A06 se trouve à l'état logique un.

La sortie de l'inverseur 01C02 est également appliquée à l'entrée d'établissement de la bascule type —D 13A06. L'état un de la bascule est détecté par le microprogramme de transmission afin d'indiquer l'information de transmission et de commande qui, en cours de transmission, se trouve stockée dans la mémoire temporaire de transmission. Cette détection est accomplie par la version TFMNE de l'instruction TAMN. La bascule 13A06 est remise à l'état zéro par la version RSTFMCB de l'instruction SC. On notera que, lorsqu'une sous-instruction est indiquée, elle peut en fait être combinée avec l'instruction de branchement, comme dans le cas de l'instruction BRSC.

Le processeur de communications poursuit alors le chargement des caractères de données dans la mémoire tampon de transmission de la fig. 6. Le processeur est capable de charger de nouvelles

données dans la mémoire uniquement lorsque les registres d'entrée de la mémoire sont vides. Le processeur répète ainsi l'opération de lecture d'état décrite plus haut et charge ensuite un caractère de donnée. L'état des registres d'entrée vides est identifié par la sortie bit 0 du multiplexeur 04B04, c'est-à-dire le bit 4 TAR/RAR à la fig. 15g. Si la sortie est à l'état logique un, les registres d'entrée sont vides. Si la sortie est à l'état logique zéro, le processeur de communications lance un signal d'attente (WAIT) afin d'attendre un moment où les registres d'entrée de la mémoire tampon seront vides. Le processeur de communications lance un signal d'attente pendant une opération de lecture d'état pendant laquelle la sortie bit 2 du décodeur 01A02 se trouve à l'état logique zéro, cet état se trouvant appliqué à la porte —ET 06B02. A ce moment, le processeur applique un état logique zéro à la broche 20 du connecteur 01B01, cet état se trouvant appliqué à une seconde entrée de la porte 04B02. La sortie de celle-ci passe ainsi à l'état logique zéro qui se trouve appliqué, par l'intermédiaire de la porte 12C06, à l'entrée de remise à l'état zéro de la bascule JK 11D06. La sortie de la porte —ET 12D06 se trouve ainsi mise à l'état logique un afin d'inhiber l'envoi d'autres signaux d'interruption au processeur de communications, et celui-ci se met alors en état d'attente.

Pendant ce temps, le microprogramme de transmission élimine les mots d'opération transmission dans la mémoire tampon de transmission, libérant ainsi les registres d'entrée de cette mémoire. Dans ce cas, la sortie registre d'entrée vide de la mémoire passe à l'état logique un qui fait apparaître un état logique un à la sortie de la porte —ET 12A06. Cet état logique un se trouve appliqué à l'entrée d'établissement de la bascule 11D06, par l'intermédiaire de la porte 11C06. La sortie de la porte 12D06 passe ainsi à l'état logique zéro, afin d'interrompre le processeur de communications. Celui-ci est ainsi informé qu'un nouveau caractère de donnée peut être chargé dans la mémoire tampon de transmission.

Le processus décrit ci-dessus se poursuit jusqu'à ce que le processeur de communications soit prêt à envoyer le dernier caractère de donnée d'un message à transmettre. A ce moment, le processeur lance une opération de chargement pendant laquelle le registre 01A06 se trouve chargé avec un mot de commande de transmission fin de colonne (fig. 15c), ce qui fait apparaître un état logique un à la sortie bit 3 du registre. Le processeur de communications lance alors une opération de chargement de la mémoire, pendant laquelle le mot stocké dans le registre 01A06 se trouve chargé dans les mémoires 01B06 et 03B06, et pendant lequel le dernier caractère de donnée du message se trouve chargé dans les mémoires 05B06 et 07B06. L'état logique un à la sortie bit 3 du registre 01A06 est détecté par le microprogramme de transmission, afin d'indiquer qu'un dernier caractère de donnée se trouve chargé dans la mémoire. En outre, le signal à la sortie bit 3 du registre 01A06 se trouve envoyé à la bascule 11D06, par l'intermédiaire de la porte 11C06, afin d'inhiber l'envoi au processeur d'autres signaux d'interruption.

La mémoire de transmission de la fig. 6 sépare le fonctionnement du processeur de communications 10 du fonctionnement de la logique de commande de transmission. Le fonctionnement de celle-ci est décrit maintenant. Ainsi qu'il apparaît de ce qui précède, le microprogramme de transmission se trouve dans la mémoire de commande de programme de transmission comprenant les six mémoires mortes programmables 01C07, 04C07, 07C07, 10C07, 07D07 et 10D07 (fig. 7). Ces mémoires sont adressées par le compteur de programme de transmission comprenant les compteurs 01A07, 04A07 et 07A07. En séquence normale, le compteur de programme de transmission se trouve incrémenté séquentiellement par le signal d'horloge. Pendant l'instruction de branchement, cependant, ce compteur peut être chargé par une information indiquant une adresse autre qu'un adresse séquentielle suivante du programme.

La logique de commande de programme comprend la bascule type —D 09A08 (fig. 8), le multiplexeur 10A08, le décodeur 09B08 et le décodeur 09C08. Le décodeur 09B08 est le décodeur d'instructions; ses sorties bits 0 à 7 correspondent au numéro octal représenté par les 3 bits dans la zone WTFLD du microprogramme de trans-

mission. Par exemple, les 3 bits de la zone WTFLD indiquent un nombre 5 octal. Ainsi, lorsqu'une instruction RCRRCR se trouve exécutée, la sortie bit 5 du décodeur passe à l'état logique zéro, tandis que les autres sorties du décodeur 09B08 restent à l'état logique un.

Le décodeur 09B08 est validé par la bascule 09A08 qui se trouve normalement à l'état un. La sortie Q de la bascule se trouve ainsi à l'état logique zéro. La bascule 09A08 peut être remise à zéro à la suite de l'exécution d'une instruction TAMN. Dans ce cas, la sortie Q de la bascule passe à l'état logique un, invalidant le décodeur 09B08. Ainsi, lorsque l'instruction suivant une instruction TAMN se trouve exécutée, les sorties du décodeur 09B08 se trouvent toutes à l'état logique un. Lorsqu'une instruction TAMN se trouve exécutée, la sortie bit 1 du décodeur 09B08 passe à l'état logique zéro, et cet état se trouve appliqué à l'entrée validation du multiplexeur 10A08. Pendant l'exécution de toutes les autres instructions, à l'exception d'une instruction TAMN, la sortie bit 1 du décodeur 09B08 se trouve à l'état logique un. La sortie D1 du multiplexeur se trouve ainsi à l'état logique un, ce qui maintient la bascule 09A08 à l'état un. La sortie Q de la bascule reste ainsi à l'état logique zéro et l'instruction suivante du microprogramme se trouve exécutée de la manière normale.

Lorsqu'une instruction TAMN se trouve exécutée, cependant, l'entrée validation du multiplexeur 10A08 se trouve à l'état logique zéro. L'entrée sélectionnée de ce multiplexeur peut faire passer la sortie D1 du multiplexeur à l'état logique zéro. A la fin de l'exécution de l'instruction TAMN, l'état logique zéro remet la bascule 09A08 à l'état initial. La sortie Q de la bascule passe alors à l'état logique un, ce qui a pour effet que l'instruction suivante du microprogramme de transmission se trouve négligée.

Le décodeur 09C08 est un décodeur de sous-ordres qui engendre des sous-ordres de commande. Lorsqu'une instruction de sous-ordres se trouve exécutée, la sortie bit 2 du décodeur 09B08 passe à un état logique zéro. D'une manière similaire, lorsqu'une instruction de branchement se trouve exécutée, la sortie bit 3 du décodeur 09B08 passe à l'état logique zéro. Les sorties bit 2 et bit 3 du décodeur sont appliquées à l'entrée de la porte 11C08 dont la sortie est appliquée à une entrée validation du décodeur 09C08. Ainsi, lorsqu'une instruction de sous-ordre ou de branchement se trouve exécutée, la sortie de la porte 11C08 passe à un état logique zéro. Si la sortie bit 3 de la mémoire morte programmable 01C07 est à un état logique zéro également, une des huit sorties du décodeur 09C08 passe à l'état logique zéro, comme indiqué par les trois entrées du décodeur. Les trois entrées du décodeur 09C08 correspondent aux trois bits constituant la zone SCFLD du microprogramme de transmission. Ainsi, par exemple, lorsque la version STCRI de l'instruction de sous-ordre de commande ou de l'instruction de branchement et de sous-ordre se trouve exécutée, un code octal de 3 se trouve appliqué à l'entrée du décodeur 09C08, et la sortie bit 8 du décodeur passe à l'état logique zéro.

Comme montré à la fig. 16, le microprocesseur de transmission comprend essentiellement trois parties, à savoir: une mémoire à accès sélectif 16 mots $\times$ 4 bits, un registre Q et une unité arithmétique et logique ALU. La mémoire à accès sélectif RAM est utilisée pour le stockage temporaire des données transmises, des instructions de commande de transmission et des états de transmission. Le registre Q est utilisé pour le stockage temporaire de diverses informations de commande. Quant à l'unité arithmétique et logique ALU, elle exécute les diverses opérations arithmétiques et logiques. Un dispositif de décalage gauche/droite comprenant les registres 120 et 122 est également prévu, en sorte que les résultats des opérations effectuées par l'unité arithmétique et logique puissent être décalés à gauche ou à droite avant d'être stockés dans la mémoire à accès sélectif RAM ou dans le registre Q.

Ainsi qu'il a été décrit plus haut, le microprocesseur est capable d'exécuter un certain nombre d'instructions de types différents déterminés par les données provenant des mémoires mortes programmables de la fig. 7. Huit lignes d'adresse provenant des mémoires de la fig. 7 sont connectées aux lignes d'adresses RAM A et aux lignes

d'adresses RAM B du microprocesseur (fig. 8), afin de désigner les adresses d'accès A et B de la mémoire à accès sélectif RAM. Les données contenues dans le microprocesseur sont fournies par l'une quelconque des six sources suivantes: la mémoire de commande 07D07, le registre 06A06, les quatre mémoires 01B06, 03B06, 05B06 et 07B06 par l'intermédiaire des deux sélecteurs 02C06 et 06C06. Le processus de lecture des données dans chacune de ces sources sera décrit plus loin. On notera que les données sont fournies au microprocesseur par l'intermédiaire de quatre circuits — ET câblés 01A08 à 07A08.

En plus des deux sorties de décalage qui ont été décrites ci-dessus, le microprocesseur 01B08 a deux sorties. La première de ces sorties est la sortie données bit 1 qui se trouve connectée à la bascule 02C08. La seconde sortie est connectée à l'entrée 6 du multiplexeur 10A08 et à l'inverseur 13A08. La sortie de cet inverseur est connectée aux entrées 7 du multiplexeur 10A08. Cette connexion avec le multiplexeur 10A08, qui est le multiplexeur d'instruction TAMN, offre au microprogramme un moyen pour vérifier les résultats de n'importe quelle opération exécutée au sein du microprocesseur 01B08. Dans un exemple de la version ALUEZ de l'instruction TAMN, le contenu de l'emplacement E de la mémoire RAM est combiné par fonction logique ET avec la constante 1 hexadécimale fournie par le programme. Si le résultat de cette opération est zéro, une instruction de branchement se trouve exécutée. Si le résultat de l'opération n'est pas zéro, l'instruction de branchement est négligée et l'instruction significative suivante sera une instruction MMP. Dans un exemple de la version ALUEZ de l'instruction TAMN, le contenu de l'emplacement 4 dans la mémoire RAM est soustrait de la donnée présente sur les lignes d'entrée données du microprocesseur 01B08, cette donnée étant dans ce cas une constante de valeur hexadécimale C. Si le résultat de cette opération a une valeur différente de zéro, l'instruction à la prochaine ligne se trouve exécutée de la manière normale. Par contre, si le résultat de l'opération est zéro, l'instruction à cette ligne est négligée et l'instruction suivante se trouve exécutée.

Le microprogramme de transmission a pour mission de gérer l'envoi des données bit par bit à l'équipement de communications de données local. Le premier type de données est constitué par les séquences code complémentaire ou abandon qui sont transmises entre les colonnes. Une séquence code complémentaire est constituée d'un bit zéro suivi de six bits un, eux-mêmes suivis d'un bit zéro, c'est-à-dire 7E hexadécimale. Une séquence d'abandon est constituée d'au moins sept bits un, c'est-à-dire FF hexadécimale. Ces séquences sont engendrées par le microprogramme de transmission sous la commande du processeur de communications, par l'intermédiaire de l'information de commande chargée dans le registre 06A06 (fig. 6).

Le second type de données est constitué par le code début de colonne qui est automatiquement engendré par le microprogramme de transmission avant l'envoi du premier caractère de données de chaque colonne. Le troisième type de données est constitué par les caractères de données de la colonne, tous ces caractères étant fournis par le processeur de communications par l'intermédiaire des mémoires 05B06 et 07B06. Le quatrième type de données est constitué par les données FCS qui sont engendrées par le microprogramme à l'aide du registre de contrôle de parité 03C08. Cette séquence de contrôle d'erreur est calculée en utilisant chaque bit de données de la colonne fourni par le processeur de communications, et elle est transmise à la fin de la colonne après le dernier caractère. Le cinquième type de données est constitué par le code fin de colonne qui se trouve automatiquement engendré par le microprogramme de transmission et envoyé après le code FCS. Le sixième type de données est constitué par la séquence d'abandon qui se trouve automatiquement engendrée et envoyée par le microprogramme chaque fois que se produit un état de non-exécution.

Un nouveau bit de données est envoyé à l'équipement de communications de données DCE local lors de l'occurrence de chaque front montant du signal d'horloge provenant de l'équipement DCE, de la manière suivante. Le signal d'horloge est reçu de l'équipement

DCE local à la broche 02 du connecteur 02D01 (fig. 1). Ce signal est appliqué au récepteur d'entrée 03A05. La sortie de ce récepteur est connectée à l'entrée du récepteur 03B05, dont la sortie est elle-même connectée à l'entrée du multiplexeur 11C04 (fig. 4). Plus particulièrement, la sortie du récepteur 03B05 est appliquée à l'entrée zéro du circuit 1 du multiplexeur. La sortie du circuit 1 du multiplexeur est appliquée aux bascules 01D08 et 02D08. Notant que le bit suivant de la donnée transmise se trouve ainsi dérivé de la bascule 02C08 comme décrit plus bas, et que la sortie Q de cette bascule est connectée à l'entrée donnée de la bascule 02D08, on voit que le bit suivant de la donnée transmise passe de la bascule 02C08 dans la bascule 02D08 lors de l'occurrence du front montant du signal d'horloge. La sortie Q de la bascule 02D08 est connectée à l'entrée du dispositif d'excitation 04C05 (fig. 5) dont la sortie est connectée à la diode de Zener 04D05, et la sortie de cette diode est connectée à la broche 6 du connecteur 02D01. La donnée transmise est aussi envoyée à l'équipement DCE local.

Comme mentionné plus haut, le signal d'horloge de transmission est également appliqué à l'entrée d'horloge de la bascule 01D08. En raison de la présence d'une résistance, qui est un niveau logique un, à l'entrée données de la bascule 01D08, la sortie Q de cette bascule passe à l'état logique un lors de l'occurrence de chaque front montant du signal d'horloge. La mise à l'état un de la bascule 01D08 est utilisée pour indiquer au microprogramme de transmission que le bit précédent transmis est passé de la bascule 02C08 à la bascule 02D08, et qu'il est dès lors temps de charger le bit suivant dans la bascule 02C08. Le microprogramme de transmission surveille la bascule 01D08 par l'intermédiaire de la version TRDY et de l'instruction TAMN. Ici, l'état un de la bascule 01D08 a pour effet que l'instruction de branchement à une ligne est négligée et que c'est l'instruction à une prochaine qui se trouve exécutée. Lors de la détection de l'état un de la bascule 01D08, le microprogramme de transmission remet cette bascule à l'état initial par l'intermédiaire de la version RSTRDY de l'instruction SC ou de l'instruction BRSC. Après la remise à l'état zéro de la bascule 01D08, le microprogramme commande l'application du bit de données suivant à la bascule 02C08, comme décrit plus bas.

A chaque intervalle de bit de transmission, un nouveau bit de données doit être chargé dans la bascule 02C08. Cela est effectué par la version WRTBB de l'instruction SC. Lorsque cette instruction est exécutée, le bit de données suivant, qui se trouve stocké dans la position binaire extrême droite ou la moins significative de l'emplacement E de la mémoire RAM, se trouve appliqué à la ligne bit 1 du microprocesseur 01B08. Ce bit est à son tour appliqué à l'entrée données de la bascule 02C08. Cette instruction SC particulière fait passer la sortie 1 du décodeur 09C08 à l'état logique zéro, puis à l'état logique un. Ce signal est appliqué à l'entrée d'horloge de la bascule 02C08 et provoque l'enregistrement du bit de données suivant dans cette bascule lors de l'exécution de cette instruction. Cette opération par laquelle un nouveau bit de données se trouve chargé dans la bascule 02C08 est appelée chargement de la mémoire tampon de bit transmission.

Pendant qu'il attend que le processeur de communications 10 délivre les données à transmettre pour la colonne, le microprogramme de transmission engendre et envoie des caractères de remplissage des séquences code complémentaire ou abandon. Cela se fait bit par bit, en chargeant le bit de données approprié dans la bascule 02C08 à chaque front montant, comme expliqué plus haut. Le choix entre séquences code complémentaire ou abandon est fait par le processeur de communications et sera expliqué plus loin.

La longueur de chaque multiplet est contrôlée par un compteur de quatre multiplets stockés dans un des 60 emplacements de la mémoire RAM du microprocesseur 01B08. Ce compteur reçoit initialement un contenu de huit par une instruction MMP qui copie les bits zone constante de la mémoire de contrôle 07D07 (fig. 7) sur les lignes d'entrée données du microprocesseur, par l'intermédiaire des circuits — ET câblée 01A08 à 07A08. A chaque intervalle de bit, ce compteur régresse d'une unité et est vérifié par la version ALUEZ de

l'instruction TAMN pour voir si une valeur zéro a été atteinte, qui indique qu'une limite de multiplet a été atteinte. L'apparition d'un état constitué d'uniquement des bits zéro est appelé ci-après limite de multiplet transmission.

A chaque limite de multiplet, le microprogramme doit engendrer et stocker le caractère de remplissage suivant, qui sera introduit dans la bascule 02C08 lors de l'intervalle de bit subséquent. Le choix entre une séquence code complémentaire ou une séquence d'abandon est déterminé par le bit 3 du mot de commande de transmission étendu de 4 bits dans le registre 06A06. Le contenu de ce registre est introduit dans le microprocesseur 01B08 par l'intermédiaire de la version TXCW de l'instruction RER. Lorsqu'une instruction RER, est exécutée, les sept sorties du décodeur 09B08 passent à l'état logique zéro qui se trouve appliqué à une entrée de la porte 05D08. Pendant l'exécution de la version TXCW de l'instruction RER, la sortie de la mémoire 01C07 qui est appliquée à la porte 05D08 est également à l'état logique zéro. La sortie de la porte 05D08 est connectée aux entrées F du registre 06A06. L'état logique zéro à la sortie de la mémoire 01C07 valide ainsi la sortie du registre 06A06. Ces sorties sont connectées aux quatre fonctions ET câblées 01A08 à 07A08, qui se trouvent à leur tour connectées à l'entrée du microprocesseur 01B08. De cette manière, le mot de commande précédemment délivré par le processeur de communications se trouve chargé dans le microprocesseur de transmission pour un usage ultérieur. On notera que, pendant l'exécution de l'instruction RER, la sortie de l'inverseur 13C08, qui est connectée à l'entrée validation sortie 08 de la mémoire 07D07, est à l'état logique un qui inhibe les sorties de cette mémoire. En d'autres termes, les données extraites de cette mémoire sont appliquées aux entrées données du microprocesseur de transmission à tout moment, sauf pendant l'exécution d'une instruction RER.

Après que le mot de commande étendu a été introduit dans le microprocesseur, comme décrit plus haut, le bit TIFM (bit 3 à la fig. 15a) est examiné, afin de déterminer si c'est une séquence code complémentaire ou une séquence d'abandon qui doit être engendrée. Le bit est vérifié par l'instruction TAMN. Une séquence code complémentaire ou d'abandon est alors engendrée et stockée dans un emplacement de la mémoire interne RAM du microprocesseur. La séquence code complémentaire ou d'abandon ainsi engendrée est envoyée bit par bit dans les positions binaires limites successives. Après l'envoi de chaque bit, le registre de la mémoire RAM est décalé d'une position à droite.

Ce processus de génération de séquences code complémentaire ou d'abandon, puis de transmission de ces séquences, se trouve répété jusqu'à ce que le processeur de communications délivre les données à transmettre pour la colonne, comme il sera décrit plus loin. On notera que le processus général décrit plus haut est le même pour toutes les données. La seule différence significative qui peut se produire dans le cas où certains types de données sont délivrés par le processeur de communications 10 est que, dans certains cas, la longueur d'un caractère peut être inférieure à 8 bits. Dans ce cas, le compteur de longueur de multiplets que contient la mémoire RAM se trouve initialisé à la valeur convenable au lieu de 8 à chaque limite de multiplet.

A chaque limite de multiplet, le programme de transmission vérifie si l'interruption précédente a été achevée. Cela se fait au moyen de la version TCRI de l'instruction TAMN. Cette instruction fournit un moyen pour vérifier la bascule d'interruption de transmission 11D06 (fig. 6). Si l'opération devant être exécutée pendant l'interruption précédente est achevée, l'instruction qui suit l'instruction TAMN se trouve exécutée. Par contre, si l'opération devant être exécutée pendant l'interruption précédente n'est pas achevée, l'instruction suivante est négligée et c'est la seconde instruction qui suit qui se trouve exécutée. Il est entendu que le processeur de communications 10 a chargé un bit un dans la position binaire 3 du registre 01A06 qui, après avoir été inversé par la porte 04A06, devient un bit zéro à l'entrée de la porte 11C06, ce signal maintenant la sortie de cette porte à l'état logique zéro et empêchant ainsi la bascule d'inter-

ruption 11D06 d'être mise à l'état un. Lorsque, à une limite de multiplet, le microprogramme de transmission détecte que l'opération devant être exécutée pendant l'interruption précédente est achevée, le microprogramme commande l'exécution de la version RSTFMCB de l'instruction SC. L'exécution de cette instruction porte la sortie bit 5 du décodeur 09B08 à l'état logique zéro. Ce signal est appliqué à l'entrée de remise à zéro du registre 01A06, ce qui remet le registre tout entier à l'état logique zéro. Il en résulte que la sortie bit 3 du décodeur se trouve portée à l'état logique zéro qui, après inversion dans l'inverseur 04A06, devient un état logique un à l'entrée de la porte 11C06. La sortie de celle-ci passe alors à l'état logique un, qui met à l'état un la bascule 11D06 et engendre un signal d'interruption pour le processeur de communications 10, de la manière décrite précédemment. Comme le processeur de communications 10 répond à ce signal d'interruption en mettant à l'état un le bit 3 du registre 01A06, bloquant ainsi les signaux d'interruption ultérieurs, on voit qu'une interruption se trouve engendrée pour chaque multiplet. Lorsque le processeur de communications est prêt à délivrer les données pour la colonne, il répond au signal d'interruption en chargeant les données dans les mémoires 05B06 et 07B06 (fig. 6), comme décrit précédemment, au lieu de charger un bit un dans la position binaire 3 du registre 01A06.

A chaque limite de multiplet entre colonnes, le microprogramme de transmission vérifie la mémoire tampon de transmission, afin de voir si les informations de commande pour les données d'une colonne ont été fournies. Lorsqu'une donnée est disponible aux sorties de la mémoire, toutes les sorties de chacune des quatre mémoires 01B06 à 07B06 sont à l'état logique un et se trouvent appliquées à la porte 11A06. La sortie de celle-ci est à l'état logique un lorsque le registre de sortie de la mémoire contient une donnée. Cela est vérifié par le microprogramme au moyen de la version TFMOR de l'instruction TAMN. Si le registre de sortie n'est pas prêt, une séquence code complémentaire ou d'abandon additionnelle se trouve engendrée et transmise, comme décrit plus haut.

Lorsque la mémoire de transmission de la fig. 6 a une donnée disponible, le microprogramme lit l'information de contrôle et la donnée à transmettre dans la mémoire à accès sélectif interne du microprocesseur de transmission. Comme le trajet de données de la logique de commande de transmission a une largeur de 4 bits seulement, le chargement des 16 bits de la donnée et de l'information de contrôle requiert quatre instructions séparées. Quatre versions de l'instruction RER sont utilisées, à savoir les versions TCWL, TCWR, TDL et TDR. Lorsque le mot de contrôle est lu, soit pendant l'exécution d'une instruction TCWL, soit pendant l'exécution d'une instruction TCWR, la sortie de la porte 06D08, connectée à l'entrée sortie validée du multiplexeur 02C06 de la fig. 6, est à l'état logique zéro, en permettant ainsi de charger dans le microprocesseur de transmission 01B08 la donnée provenant de la mémoire 01B06 ou de la mémoire 03B06, par l'intermédiaire de quatre circuits — ET câblés 01A08, 03A08, 05A08 et 07A08.

De plus, pendant l'exécution de l'instruction TCWL, l'entrée sélection du multiplexeur 02C06 se trouve à l'état logique zéro, sélectionnant ainsi les entrées bit 0 qui sont connectées à la moitié gauche du mot de contrôle, à savoir la mémoire 01B06. Lorsqu'une version TCWL de l'instruction RER est exécutée, l'entrée sélection du multiplexeur 02C06 est à l'état logique un, validant ainsi les entrées bit 1 qui sont connectées à la mémoire 03B06. Lorsqu'une donnée est lue dans la mémoire, par l'exécution de la version TDL ou TDR de l'instruction RER par exemple, la sortie de la porte 07D08, qui est connectée au sélecteur 06C06, est à l'état logique zéro, ce qui a pour effet de transférer les données de la mémoire 05B06 ou 07B06 dans le microprocesseur de transmission. En outre, lorsque l'instruction TDL est exécutée, l'entrée sélection du multiplexeur 06C06 est à l'état logique zéro, ce qui a pour effet de sélectionner les entrées bit 0 connectées à la mémoire 05B06. Lorsque l'instruction TDR est exécutée, l'entrée sélection du multiplexeur 06C06 est à l'état logique un et ce sont les entrées bit 1 qui sont validées.

Après que l'information de contrôle et la donnée ont été introduites dans le microprocesseur, une impulsion de pointage de sortie doit être lancée vers la mémoire tampon de transmission, afin de remettre les registres de sortie à l'état initial et de permettre aux mots suivants d'être introduits dans les registres de sortie. Cela est réalisé au moyen de la version RSTFMOR de l'instruction SC. Lors de l'exécution de cette instruction, la sortie 2 du décodeur 09B08 et la sortie 6 du décodeur 09C08 passent à l'état logique zéro, de manière à mettre la bascule 09D08 à l'état un. La sortie Q de cette bascule, qui est appliquée aux quatre mémoires tampons de transmission, passe à l'état logique un et remet ainsi à l'état initial les registres de sortie des quatre mémoires. Cette bascule se trouve remise à l'état zéro dès qu'une autre instruction se trouve exécutée. Après cette opération, une nouvelle information de donnée de contrôle peut être introduite dans les registres de sortie des mémoires tampons.

Le microprogramme de transmission utilise l'instruction TAMN pour examiner le mot de contrôle associé à chaque caractère de donnée transmis. Le bit TEOF (bit 3 à la fig. 15c) est le bit de contrôle fin de colonne, et il se trouve toujours à l'état zéro jusqu'à ce que le processeur de communications 10 ait délivré le dernier caractère de la colonne. D'autres bits du mot de contrôle sont utilisés pour désigner le format de la colonne. Les paramètres qui sont désignés indiquent si la colonne (fig. 15h) contient ou non une zone adresse étendue, une zone contrôle étendue, une zone contrôle logique ou un multiplet de contrôle essai. En outre, trois bits désignent la longueur du multiplet de la partie zone test de la colonne. Ces bits sont utilisés lorsque la colonne ne doit pas avoir de multiplet de contrôle d'essai car, dans ce cas, l'information de longueur de multiplet est comprise dans la colonne elle-même et se trouve extraite au moment voulu par le microprogramme, et se trouve utilisée ensuite au moment voulu pour contrôler la longueur de chaque multiplet. On soulignera que la disponibilité de ces fonctions de contrôle et le traitement effectué par le microprogramme réduisent considérablement les exigences imposées au processeur de communications. De cette manière, il est possible de réaliser un plus grand système. On notera également que, après que le processeur de communications a délivré le mot de contrôle initial, il n'est pas nécessaire de délivrer d'autres informations de contrôle, jusqu'à ce que le dernier caractère de la colonne ait été délivré.

Après que le microprogramme de transmission a détecté la présence du premier caractère d'une colonne, il commande la génération et la transmission d'une séquence code complémentaire d'ouverture avant la transmission du premier caractère. Il commande également l'initialisation de la zone CRC de l'instruction SC. L'exécution de cette instruction fait passer à l'état logique zéro la sortie bit 7 du décodeur 09C08 qui se trouve connectée à l'entrée d'établissement du générateur de contrôle de parité 03C08, de manière à mettre ce générateur dans l'état où il engendre uniquement des bits un.

A mesure que la colonne est transmise, chaque bit entre le code complémentaire d'ouverture et de fermeture se trouve appliqué au registre CRC par l'intermédiaire de l'instruction UDSCR. Lorsque cette instruction est exécutée, la sortie 5 du décodeur 09B08 reste à l'état logique un, mais la sortie 6 du décodeur passe à l'état logique zéro. Ces signaux sont appliqués à la porte 07C08 par l'intermédiaire de la porte 07B08 et, de là, à l'entrée de validation du générateur 03C08. L'entrée d'horloge de la porte 07C08 fait apparaître un état logique un à sa sortie qui se trouve connectée à l'entrée d'horloge du générateur 03C08. On notera que la sortie de la bascule 02C08 est connectée à l'entrée données du générateur 03C08 et, comme l'entrée de validation est à l'état logique un, comme indiqué plus haut, l'entrée d'horloge provoque l'introduction du nouveau bit de donnée dans le générateur.

Le microprogramme de transmission surveille le flot de données à mesure qu'il se trouve envoyé à l'équipement de communications de données DCE local, et il engendre, puis infère un bit de donnée zéro chaque fois que se produisent en chaîne cinq bits un contigus,

comme exigé par le protocole de commande de transmission de données à haut niveau.

Après que le programme de transmission a achevé l'envoi du code complémentaire d'ouverture de la colonne, il commande l'envoi du premier caractère de donnée qui a été obtenu de la mémoire tampon de transmission, comme décrit plus haut. Après que ce premier caractère a été transmis, le microprogramme commande à nouveau la vérification du registre de sortie de la mémoire tampon, par l'intermédiaire de la version TFMOR de l'instruction TAMN, afin de déterminer si un autre caractère est disponible. Dans l'affirmative, le programme commande la lecture du mot de contrôle et du caractère de donnée pour l'introduire dans la mémoire à accès sélectif interne du microprocesseur de transmission, comme décrit plus haut. Supposant que le bit 3 du mot de contrôle (bit indicateur de fin de colonne) passe à l'état un, le programme commande la transmission d'un second caractère, de la manière décrite plus haut. Ce processus se répète pour chaque caractère de la colonne.

Si, en cours de transmission d'une colonne, comme décrit plus haut, il est constaté que la sortie de la mémoire tampon n'est pas prête, on dit qu'il s'est produit un état de non-exécution ou d'exécution incomplète. Dans ce cas, le programme commande l'exécution de la version STUNDRN de l'instruction SC, qui a pour effet de mettre à l'état un la bascule 06A04. Ensuite, lorsque le processeur de communications 10 lit l'état final à la fin de la colonne, il se trouvera avisé de l'état d'exécution incomplète par l'intermédiaire du signal TUNDRN appliqué au quatrième bit du multiplexeur 04B04 par la bascule 06A04. Pendant ce temps, le microprogramme de transmission commande la génération d'une séquence d'abandon et son envoi bit par bit à l'équipement DCE local. Une colonne ainsi abandonnée indique au terminal distant qu'une erreur s'est produite au terminal de départ. Après que la séquence d'abandon a été transmise, le microprogramme continue à engendrer et à envoyer des séquences code complémentaire tout en attendant un état fin de colonne et l'achèvement des opérations de toutes les interruptions antérieures. Cette particularité de l'adaptateur de lignes d'envoyer une séquence d'abandon seulement, suivie de séquences code complémentaire, est une fonction importante car elle abandonne correctement le message tout en ne laissant pas la ligne de transmission en état d'occupation. On notera que cela est exécuté entièrement par la logique de transmission de l'adaptateur de lignes, sans nécessiter aucune intervention du processeur de communications déjà surchargé.

Chaque caractère de la colonne est dérivé de la mémoire tampon par le microprogramme de transmission et envoyé bit par bit à l'équipement de communications de données DCE local, comme décrit plus haut. Chaque caractère est surveillé comme il convient, principalement dans le but de contrôler la longueur de multiplet de la zone texte du message. Par exemple, si la colonne contient un multiplet de contrôle d'essai TCB (fig. 15h), le programme surveille la colonne d'abord dans les multiplets d'adresse, puis dans les multiplets de contrôle. Ensuite, lors de la réception du multiplet de contrôle d'essai, l'information relative à la longueur de multiplet est extraite de ce multiplet et stockée dans la mémoire à accès sélectif interne du microprocesseur. Le programme surveille ensuite la zone de contrôle logique LCF, si besoin est, puis, lors de la réception du premier multiplet de contrôle de texte de la colonne, il commute la dimension de multiplets de 8 bits (qui s'applique aux multiplets d'adresse, de contrôle, de contrôle de test et à la zone contrôle logique de la colonne) dans la longueur spécifiée par le multiplet de contrôle d'essai. Ce processus se répète aussi longtemps que le bit indicateur de fin de colonne du mot de contrôle est trouvé à l'état zéro.

Le processeur de communications 10 peut également spécifier des changements de format à l'aide du mot de contrôle de transmission (fig. 15b). Le bit TCFX (bit 0) du mot de contrôle, lorsqu'il est à l'état un, indique la présence dans la colonne d'une zone de contrôle étendue. Le bit TAFX (bit 1), lorsqu'il est à l'état un, indique la présence d'une zone adresse étendue. Le bit TTCB (bit 2), lorsqu'il est à

l'état un, indique la présence dans la colonne d'un multiplet de contrôle de texte TCB (fig. 15h) et le bit TLCF (bit 7), lorsqu'il est à l'état un, indique la présence de la zone contrôle logique LCF. Les bits TBS (bits 4, 5 et 6) spécifient la longueur de multiplet de la zone texte de la colonne. Le bit TEOF (bit 3) est normalement à l'état zéro et ne se trouve mis à l'état un que pour indiquer le dernier caractère de la colonne. Ces bits de contrôle dans le mot de contrôle sont transférés au registre 10A06 par l'intermédiaire des mémoires 10B06 et 03B06, et ils sont lus et manipulés sous la commande du microprogramme de transmission.

Le processeur de communications peut spécifier un mode d'opération dit mode flot de bits en introduisant le code 011 dans les positions binaires TBS. Ce mode d'opération assure la transmission de données à partir, disons, d'une mémoire de l'ensemble ordinateur dans une autre mémoire dans laquelle les données ne suivent aucune structure de caractère particulière (par exemple pendant un vidage de mémoire). Dans ce cas, il est souhaitable d'envoyer les multiplets de données de 8 bits d'ordre supérieur, ou le bit le moins significatif d'abord. Lorsque ce mode d'opération est spécifié, soit directement par le processeur de communications, par l'intermédiaire du mot de contrôle, soit indirectement par le contenu du multiplet de contrôle de texte dans le flot de données lui-même, le microprogramme de commande pousse chaque caractère avant de le transmettre à l'équipement DCE local.

Une autre particularité significative de l'adaptateur, qui implique l'utilisation de mots de contrôle produits par le processeur de communications, réside dans l'utilisation du bit TBFM dans le mot de contrôle étendu (fig. 15a). Normalement, dès qu'une donnée est chargée dans la mémoire tampon, atteint la sortie de cette mémoire et se trouve détectée par le microprogramme de transmission, celui-ci commande l'envoi d'un code complémentaire d'ouverture, puis la transmission de la colonne. Cela n'est pas toujours souhaitable, car le processeur de communications doit continuer à délivrer des caractères à cadence rapide, afin de travailler de pair avec la ligne de transmission à vitesse élevée ou risquer l'apparition d'une erreur d'exécution incomplète.

Toutefois, lorsque le processeur de communications met le bit TBFM à l'état un dans le mot de contrôle stocké dans le registre 06A06, le microprogramme de transmission détecte cet état et envoie des séquences code complémentaire, mais ne commande ni la suppression, ni l'envoi de données réelles extraites de la mémoire tampon. Ce mode de fonctionnement se poursuit aussi longtemps que le bit TBFM est à l'état un. Le processeur de communications peut alors charger jusqu'à 64 caractères dans la mémoire tampon, à une cadence convenant à la situation qui prévaut. Cela revient à dire que, lorsque le processeur de communications a d'autres opérations à exécuter et ne dessert pas uniquement l'adaptateur de lignes BHCLA, il peut charger la mémoire tampon de transmission avec un ordre de priorité faible, sans craindre un état d'exécution incomplète. Inversement, lorsque le processeur de communications peut s'accommoder d'une vitesse de transfert de données élevée, il met le bit TBFM à l'état zéro afin de permettre au microprogramme de commander le démarrage de la transmission de la colonne immédiatement.

Lorsque le bit fin de colonne TEOF est à l'état un dans le mot de contrôle (fig. 15c), le dernier caractère de donnée du message est présent. La première action pour mettre fin à une colonne est la transmission du dernier caractère de donnée. Cette transmission est semblable à celle de tous les autres caractères qui précèdent, sauf que ce multiplet peut être un multiplet partiel, ce qu'indique l'état du bit TPLB, et que la longueur du caractère dans ce multiplet peut aller de 1 à 8 bits, ce que déterminent les bits TLBS (bits 4, 5 et 6 du mot de contrôle fin de colonne).

L'action suivante à la fin d'une colonne est la transmission de la séquence CRC de 16 bits qui a été calculée pour la colonne et se trouve dans le générateur 03C08. Les huit premiers bits sont introduits dans le microprocesseur bit par bit, ils y sont complétés, stockés dans la mémoire à accès sélectif RAM interne, puis envoyés

bit par bit à l'équipement DCE local, de la même manière que les autres caractères de données. Ce processus se répète pour les huit bits restants. La lecture réelle du contenu du registre et son introduction dans le microprocesseur sont effectuées à l'aide de l'instruction

5 RCRCR.

Cette instruction est semblable à l'instruction UCRCR décrite précédemment sauf que, dans ce cas, la sortie bit 5 du décodeur 09B08, qui est connectée à l'entrée validation du générateur 03C08, passe à l'état logique zéro, ce qui permet à la séquence CRC calculée d'être extraite au lieu de provoquer l'établissement d'un nouveau calcul. Après que la séquence CRC complétée (appelée séquence FCS dans le protocole de commande de transmission de données à haut niveau) a été transmise, une séquence code complémentaire de fermeture se trouve engendrée, stockée et envoyée bit par bit sous la direction du programme. Un signal d'interruption final est ensuite engendré et envoyé au processeur de communications, comme décrit plus loin.

Ce qui précède décrit l'opération fin de colonne normale. La logique de commande de transmission permet cependant un certain nombre de variantes de ce processus normal. Ces variantes sont sélectionnées par le processeur de communications 10 et indiquées à la logique de commande de transmission au moyen du mot de contrôle de transmission associé au dernier caractère de donnée transmis (fig. 15c). Après que le programme a commandé la lecture du dernier caractère de donnée et du mot de contrôle associé dans la mémoire tampon de transmission, il commande l'examen de ces bits afin de voir si une opération spéciale est requise. Tout d'abord, le bit TNFCS spécifie si la séquence CRC calculée doit être envoyée ou non. L'utilisateur peut souhaiter engendrer sa propre séquence de vérification, auquel cas cette séquence peut être communiquée à sa logique de commande de transmission comme partie du flot de données à transmettre. En second lieu, l'utilisateur peut souhaiter terminer la colonne par une séquence d'abandon au lieu d'une séquence code complémentaire, afin d'établir un processus d'abandon de colonne. Une telle opération est effectuée lorsque le bit TA se trouve à l'état un. Troisièmement, un procédé est prévu en sorte que la logique de réception puisse recevoir un ordre de resynchronisation du transmetteur dès que la transmission de la colonne s'achève. Cette opération est effectuée lorsque le bit TRR est à l'état logique un. Cette opération permet à l'utilisateur de resynchroniser le récepteur dès que le transmetteur a achevé la transmission de la colonne en prévision d'un message à renvoyer au récepteur par le transmetteur du terminal distant. Cette resynchronisation du récepteur se fait au moyen de la version SRCRSYN de l'instruction SC.

A la fin de la colonne, le microprogramme de transmission commande la génération d'un signal d'interruption pour le processeur de communications. Cela est réalisé au moyen de la version STCRI de l'instruction SC. L'exécution de cette instruction fait passer momentanément à l'état logique zéro la sortie bit 3 du décodeur 09C08, ce qui a pour effet de mettre la bascule 11D06 à l'état un et d'engendrer un signal d'interruption, comme décrit précédemment. On notera que le signal d'interruption est engendré même lorsque toutes les entrées de la porte 11C06 ne sont pas à l'état logique un. Plus particulièrement, même si le bit prêt à envoyer, reçu de l'équipement DCE local, est à l'état zéro, le signal d'interruption se trouve engendré. En réponse à ce signal, le processeur de communications lit l'état de l'équipement DCE de la manière décrite précédemment.

A ce moment, le processeur de communications examine la sortie bit 4 du multiplexeur 04B04 qui se trouve connectée à la broche 9 du connecteur 01C01 par l'intermédiaire du nœud circuit — ET câblé 08D04. L'état de la broche 9 du connecteur 01C01 reflète l'état de la bascule 06A04 à ce moment, ce qui permet de déterminer si la colonne a été transmise correctement ou si une erreur d'exécution incomplète s'est produite. Le processeur de communications exécute alors une opération d'attente qui porte à l'état logique zéro la sortie de la porte 12C06. Cette sortie se trouve connectée à la bascule 11D06 et à la bascule exécution incomplète 06A04, de sorte que ces

deux bascules se trouvent remises à l'état zéro. Cela a pour effet de supprimer l'état d'interruption.

Le microprogramme de commande de transmission est à ce moment à nouveau à l'état intercolonne. Comme indiqué précédemment, des séquences code complémentaire ou d'abandon sont transmises dans cet état, dans l'attente du premier caractère de donnée d'une nouvelle colonne en provenance du processeur de communications. Un signal d'interruption est normalement engendré et envoyé au processeur de communications 10 à chaque limite de multiplet. Si le processeur décide de mettre fin aux opérations de transmission au lieu d'envoyer une autre colonne, ces signaux d'interruption engendrés à chaque limite de multiplet peuvent être utilisés comme signal de temps. Plus particulièrement, le processeur de communications peut attendre l'apparition de trois signaux d'interruption, par exemple, avant de remettre à l'état initial la ligne demande d'envoi allant vers l'équipement DCE local. Cela assure que le message qui vient d'être transmis a été complètement reçu par le récepteur au terminal de données distant.

Opération de réception

L'opération de réception commence lorsque le processeur de communications charge un mot de commande réception (fig. 15d) dans les registres 01A10 et 05A10 (fig. 10). Ces deux registres sont appelés dans ce qui suit registre de mot de commande réception. En chargeant ce registre, le processeur de communications applique d'abord l'information de commande appropriée aux huit lignes CPDAT. Le processeur applique alors un signal aux six lignes d'adresse et de contrôle aboutissant au décodeur d'adresse réception 04A02, afin que celui-ci porte à l'état logique zéro sa sortie bit 1 qui se trouve connectée à une entrée de validation du registre de mot de commande réception. Le processeur envoie ensuite un signal de pointage qui se trouve appliqué aux entrées d'horloge du registre de mot de commande réception, et il envoie l'information de commande dans le registre de mot de commande.

Le processeur de communications 10 initialise alors la logique de commande de réception en commandant l'exécution d'une opération de resynchronisation de réception. Le processeur envoie un signal sur les six lignes d'adresse et de commande aboutissant au décodeur d'adresse réception 04A02, en sorte que la sortie bit 4 de celui-ci, qui se trouve connectée à la porte 07B02, passe à l'état logique zéro. La sortie de la porte 07B02 est appliquée à une entrée de la porte 07C02 qui reçoit, à une seconde entrée, un signal de pointage provenant du processeur de communications, de telle sorte que la sortie de cette porte produit une impulsion à front positif. Ce signal est à son tour appliqué à la bascule 07D02, afin d'établir celle-ci à l'état un. La sortie de cette bascule est détectée par la logique de commande réception, comme il sera décrit plus loin. Cette opération de resynchronisation peut être effectuée à tout moment et elle peut également être effectuée d'une manière similaire par le processeur de communications, par l'intermédiaire du décodeur d'adresse de transmission 01A02.

La mise au point initiale de la partie récepteur de l'adaptateur de ligne 14 par le processeur de communications 10 se trouve ainsi effectuée. La logique de commande de réception et le programme de commande de réception sont décrits à présent pour montrer comment sont fournies les données de réception et les informations d'état de réception pour le processeur de communications 10. On montrera ensuite comment sont engendrés les signaux d'interruption envoyés au processeur de communications, et comment celui-ci répond à ces signaux d'interruption en lisant le caractère de donnée reçu et le mot d'état associé. Toute référence au microprocesseur sous-entend la paire de microprocesseurs de réception 01C10 et 07C10 de la fig. 10, qui sont décrits plus loin, sauf indication contraire. D'une manière similaire, toute référence à la mémoire premier-entré premier-sorti désigne la combinaison des quatre mémoires 01B12, 03B12, 06B12 et 08B12, qui sont décrites plus loin, sauf indication contraire.

La logique de commande de réception comprend fondamentalement quatre parties: le compteur d'adresse de programme, la mémoire de commande de programme, la logique d'exécution d'instruction et la paire de microprocesseurs. Le compteur d'adresse de programme comprend les registres 01A09, 04A09 et 07A09 dont le fonctionnement a été décrit précédemment. La mémoire de commande de programme comprend les mémoires mortes programmables 01C09, 04C09, 07C09, 10C09, 04D09, 07D09 et 10D09 dont les fonctions ont été décrites précédemment. La logique d'exécution d'instructions comprend le décodeur d'instruction de base 06A11 (fig. 11), le décodeur générateur de sous-instruction 08C11, le multiplexeur 09A11 et la bascule 08A11. Chacune de ces fonctions a été partiellement décrite dans ce qui précède. Le fonctionnement de ces éléments est virtuellement identique au fonctionnement des éléments équivalents de la logique de commande de transmission (c'est-à-dire, respectivement, le fonctionnement du décodeur 09B08, du décodeur 09C08, du multiplexeur 10A08 et de la bascule 09A08).

Le microprocesseur de réception 8 bits comprend le microprocesseur 4 bits 01C10 et le microprocesseur 4 bits 07C10, qui ont également été décrits précédemment. Plus particulièrement, le microprocesseur de transmission a été décrit et, ensuite, on a décrit les similitudes et différences du microprocesseur de réception par rapport au microprocesseur de transmission. Les huit lignes d'entrée de données du microprocesseur reçoivent un multiplet de données de 8 bits des registres 01A10 à 05A10 pendant les instructions AR, et des mémoires mortes 04D09 et 07D09 pendant toutes les autres instructions. Les huit lignes de sortie de données du microprocesseur sont connectées à la mémoire 04A12 où les multiplets de données et d'état sont temporairement stockés avant d'être chargés dans la partie appropriée de la mémoire tampon premier-entré premier-sorti (fig. 12a). En outre, le bit le plus significatif du processeur 01C10, c'est-à-dire le bit zéro, se trouve appliqué à l'entrée données du générateur de contrôle de parité 04C11. Les sorties des deux microprocesseurs 4 bits sont combinées par fonction — ET au nœud 06D10 et appliquées au multiplexeur 09A11. Ce signal est également appliqué à l'inverseur 13A11 dont la sortie est connectée à une autre entrée du multiplexeur 09A11. On notera que les signaux de sortie-ALU-égale-zéro des deux microprocesseurs doivent être vrais, comme indiqué par l'état logique un, avant que le signal composite n'atteigne le niveau logique un qui indique que la sortie totale des deux microprocesseurs est en fait zéro. Les broches de décalage entrée/sortie sont connectées comme décrit précédemment.

Le microprogramme de commande de réception surveille la bascule de resynchronisation 07D02 qui peut être mise à l'état un soit directement par le processeur de communications, soit par le programme de commande de transmission, comme décrit plus haut, afin de vérifier si un ordre de resynchronisation a été reçu. La vérification de l'état de cette bascule est effectuée par l'intermédiaire de la version RCRSYN de l'instruction TAMN. Si la bascule 07D02 est à l'état zéro, un branchement à la ligne prochaine est négligé et le programme passe alors à l'exécution d'une autre instruction. Par contre, si la bascule 07D02 est à l'état un, le branchement indiqué se trouve exécuté et le programme se poursuit par là. L'initialisation interne nécessaire pour la synchronisation est alors effectuée et une instruction RSRCSYN se trouve exécutée, afin de remettre la bascule de resynchronisation à l'état zéro.

Chaque bit de données est reçu de l'équipement de communications de données DCE local à la broche 14 du connecteur 02D01 (fig. 1). Ce signal est acheminé vers le récepteur 05B05 par l'intermédiaire du récepteur 05A05 puis, de là, il est acheminé vers le multiplexeur 11C04 et, enfin, vers l'entrée données de la bascule 02C11. Le signal d'horloge est reçu de l'équipement DCE à la broche 10 du connecteur 02D01. Ce signal est acheminé à travers le récepteur 04A05, le récepteur 04B05, le multiplexeur 11C04 et l'inverseur 12D04 jusqu'aux entrées d'horloge des bascules 01C11 et 02C11 (fig. 11). Ces deux bascules sont alors déclenchées par les fronts négatifs du signal d'horloge. Dans le cas de la bascule 02C11, le bit de données suivant, indiqué plus haut, est ainsi chargé dans la bascule.

Dans le cas de la bascule 01C11, celle-ci est mise à l'état un, afin d'indiquer la présence d'un nouveau bit de données dans la bascule 02C11. Le programme commande la vérification de cet état, par l'intermédiaire de la version RRDY de l'instruction test. Lorsque la bascule 01C11 est trouvée à l'état un, avant de commander le traitement du nouveau bit, le programme commande la remise à l'état zéro de cette bascule, par l'intermédiaire de l'instruction RSRRDY. Comme indiqué précédemment, cette instruction peut être combinée avec l'instruction de branchement pour former une instruction BRSC.

Ayant déterminé qu'un nouveau bit de données est disponible pour être traité, le programme de commande de réception introduit le bit suivant dans le microprocesseur, à l'aide de la version SRQF de l'instruction MMP. Lorsque cette instruction est exécutée, l'entrée de l'inverseur 03B11 est à l'état logique zéro et sa sortie, qui est connectée à la porte — ET 03C11, se trouve donc à l'état logique un. L'autre entrée de la porte 03C11 est également à l'état logique un à ce moment, et sa sortie est donc à l'état logique un. La sortie de la porte 03C11 est connectée à une entrée de la porte 02D11 dont une autre entrée est connectée à la sortie de la bascule 02C11. Le signal de sortie de cette bascule est donc alors transféré, par l'intermédiaire de la porte 02D11, et appliqué à l'entrée décalage à gauche du registre Q du microprocesseur 01C10.

Notant que les bits 7 et 8 appliqués au microprocesseur désignent une opération de décalage à droite à ce moment, le bit de données reçu est introduit dans le registre Q du microprocesseur. Plus particulièrement, l'opération de décalage à droite est exécutée, car la sortie bit 3 de la mémoire morte programmable 10C09 est à l'état logique un et la sortie bit 2 à l'état logique zéro. De cette manière, chaque nouveau bit de données reçu est introduit dans le microprocesseur. Cette opération est appelée dans ce qui suit opération d'introduction du bit de données suivant.

Lorsque commencent les opérations de réception, chaque bit de données est introduit dans le microprocesseur, comme il vient d'être décrit. Chaque nouveau bit est stocké dans la mémoire à accès sélectif interne du microprocesseur en même temps que les sept bits précédents. Les huit bits sont examinés bit par bit afin de voir s'ils comprennent une séquence 0111110. La détection d'une séquence code complémentaire indique que la synchronisation de caractère a été exécutée. Cette séquence code complémentaire ainsi que toute autre séquence code complémentaire sont négligées par le programme de réception pendant que celui-ci commande la recherche du premier caractère qui n'est ni un code complémentaire, ni un code d'abandon (1111111). Le premier caractère de ce genre qui est trouvé est le premier caractère d'une colonne.

Le programme commande alors la lecture des registres 01A10 et 05A10 à divers moments, afin de déterminer le format de la colonne, tel que spécifié par le processeur de communications 10. Cette opération est exécutée par l'instruction RER. L'exécution de cette instruction fait passer la sortie bit 7 du décodeur 06A11 à l'état logique zéro. Ce signal est appliqué à l'inverseur 09B11 dont la sortie est connectée aux mémoires mortes programmables 04D09 et 07D09. Celles-ci se trouvent ainsi validées et leurs sorties se trouvent mises dans un état haute impédance. On notera que, pendant l'exécution de toutes les autres instructions, les sorties de ces mémoires mortes programmables, qui contiennent la zone constante de l'instruction, sont validées. La sortie bit 7 du décodeur 06A11, qui est à l'état logique zéro, est également appliquée aux entrées sortie validée des registres 01A10 et 05A10. Le mot de commande de réception est ainsi transféré, par l'intermédiaire des nœuds 01B10 et 08B10, vers les entrées de données des microprocesseurs. On notera que les registres de mot de commande sont inhibés pendant toutes les autres instructions.

L'information de commande obtenue par l'instruction RER est examinée (si requis) par l'instruction TAMN, afin de déterminer le format de la colonne. L'information de commande fournie par le processeur de communications 10 spécifie le format de la colonne et comprend les informations concernant la zone adresse (bit RAFX),

la zone contrôle (bit RCFX), la zone commande logique (bit RLCF), le multiplet contrôle texte (bit RTCB) et la longueur de multiplet (bits RBS) pour la zone texte de la colonne (fig. 15d). Le programme de réception commande alors le traitement de chaque colonne, comme indiqué par les bits de contrôle, d'une manière similaire à celle de l'opération de transmission décrite précédemment. Un bit de contrôle additionnel RFCSIE tient compte de l'introduction de la séquence FCS, plutôt que du cas normal où la séquence CRC est vérifiée entièrement par l'adaptateur de ligne 14. Des précisions complémentaires concernant tout cela sont données ci-après.

L'adaptateur de ligne 14 assure le contrôle d'erreur de chaque message, par l'intermédiaire d'une séquence CRC. La logique comprend un circuit de contrôle d'erreur 04C11 et des circuits 06B11, 07B11, 06C11 et 04D11 (fig. 11). Au début de chaque colonne, la sous-instruction PSRFCSR se trouve exécutée afin d'initialiser le générateur de contrôle d'erreur 04C11 et de le mettre dans l'état où il produit uniquement des bits un. L'exécution de cette sous-instruction fait passer momentanément à l'état logique zéro la sortie bit 7 du décodeur 08C11, qui se trouve connectée à l'entrée d'établissement du générateur 04C11, et celui-ci engendre alors uniquement des bits un. Ensuite, pendant que chaque bit de message est reçu, une instruction FUCSR se trouve exécutée afin de charger le nouveau bit dans le générateur 04C11. Pendant l'exécution de cette instruction, le bit de données qui doit être appliqué au générateur 04C11 lui est transféré par l'intermédiaire de la sortie bit 0 du microprocesseur 01C10. La sortie bit 6 du décodeur 06A11 passe à l'état logique zéro et cet état se trouve transféré par les portes 06B11 à 06C11, afin d'appliquer l'impulsion d'horloge suivante à l'entrée d'horloge du générateur 04C11 par l'intermédiaire de la porte 06C11. Comme la sortie bit 5 du décodeur d'instructions 06A11, qui est connectée à l'entrée sortie validée du générateur 04C11, est à l'état logique pendant l'exécution de cette instruction, ce générateur de contrôle d'erreur est mis à jour au lieu de transmettre sa séquence par décalage.

Après que le premier caractère de donnée a été reçu, les bits sont introduits dans la mémoire à accès sélectif interne du microprocesseur, comme décrit plus haut. Comme les colonnes de moins de 32 bits sont illégales ou invalides, un décalage, un stockage et un retard complémentaires sont nécessaires pour permettre au programme d'écarter complètement toute colonne courte sans transfert de données ou d'informations d'état quelconque au processeur de communications. Quatre des emplacements de 8 bits de la mémoire à accès sélectif sont combinés pour former un registre à décalage de 32 bits, ces emplacements étant décalés. Tous bits de données zéro insérés par le transmetteur distant à des fins de transparence (tel que spécifié par le protocole de commande de transmission de données à haut niveau) sont éliminés du message par la logique de commande.

La mémoire tampon premier-entré premier-sorti comprend quatre circuits mémoires 01B12, 03B12, 06B12 et 08B12, ayant chacun une capacité de seize mots de 4 bits. Ces circuits sont connectés en cascade pour former une mémoire de soixante-quatre mots de 16 bits. Les circuits mémoires 01B12 et 03B12 servent à stocker l'information d'état, tandis que les circuits 06B12 et 08B12 servent à stocker les données reçues. Les informations sont copiées dans chaque paire de circuits mémoires, en provenance du registre 04A12 dont huit sorties sont connectées aux entrées données des deux paires de circuits mémoires (fig. 12a). Le registre 04A12 est à son tour chargé directement à partir du microprocesseur de 8 bits.

Après que les 32 bits de la colonne ont été reçus, indiquant ainsi une colonne valable, le premier caractère du message se trouve chargé dans la mémoire tampon premier-entré premier-sorti, de la manière suivante. Le programme commande d'abord la vérification des circuits mémoires, afin de voir si leurs registres d'entrée sont vides. L'état prêt du registre d'entrée de chaque circuit mémoire est indiqué par un bit un à sa sortie vide. Les sorties vides de chacun des quatre circuits mémoires 01B12, 03B12, 06B12 et 08B12 sont combinés dans la porte — ET 12B12. La sortie de cette porte passe à l'état logique un lorsque les circuits mémoires sont disponibles. Ce signal,

et par conséquent la disponibilité des circuits mémoires, est testé sous la direction de la version RFMIR de l'instruction d'essai.

Comme on le verra plus loin, les données et les informations d'état chargées dans la mémoire sont normalement extraites périodiquement par le bas de la mémoire par le processeur de communications et, dès lors, les registres d'entrée sont normalement vides. Dans le cas où les registres d'entrée ne sont pas disponibles soit à ce moment, soit à un moment quelconque dans le processus de traitement de la colonne, il existe un état dit de surcharge. Dans ce cas, le programme de commande attend jusqu'à ce qu'apparaisse la fin de colonne et que les registres d'entrée soient vides, puis il charge un mot d'état avec un bit un pour indiquer l'état de surcharge dans la partie état de la mémoire. Un autre bit du mot d'état est mis à l'état un pour indiquer la fin de colonne, et le troisième bit est mis à l'état un si la colonne est abandonnée, comme il sera décrit plus loin. Le processeur de communications 10 est ainsi informé de l'erreur de surcharge, et aucune tentative n'est faite pour transférer de nouvelles données de la colonne en cours au processeur, car de telles données complémentaires seraient sans signification et ne feraient qu'augmenter la charge de traitement du processeur déjà encombré.

Après que le programme a permis de déterminer que les registres d'entrée de la mémoire tampon sont disponibles, il commande le chargement du caractère de donnée reçu et du mot d'état associé dans la mémoire. Il s'agit d'un processus en deux étapes pour le mot d'état et la donnée. La donnée ou l'état est d'abord chargé dans le registre 04A12, puis copié dans la paire appropriée de circuits mémoires.

Le mot d'état associé à chaque caractère de donnée contient uniquement des bits zéro. Ce mot est engendré sous la commande de l'instruction RSRFMB. Lorsque cette instruction est exécutée, la sortie bit 4 du décodeur 08C11, qui est connectée à l'entrée de remise à zéro du registre 04A12, passe à l'état logique zéro qui remet ledit registre à l'état zéro.

Ce mot d'état contenant uniquement des bits zéro est transféré dans les circuits mémoires 01B12 et 03B12 sous la commande de l'instruction WRRFMS. Lorsque cette instruction est exécutée, la sortie bit 2 du décodeur 08C11, qui est connectée à l'entrée de mise à l'état un de la bascule 08D11, passe à l'état logique zéro, ce qui a pour effet de mettre ladite bascule à l'état un dans lequel sa sortie Q passe à l'état logique un. Lors de l'exécution de l'instruction suivante, l'entrée CK de cette bascule se trouve à l'état logique un, ce qui a pour effet de ramener la sortie Q à l'état logique zéro lors de l'occurrence de l'impulsion d'horloge suivante. Cette séquence de changement d'états logiques zéro-un-zéro à la sortie Q de la bascule 08D11, laquelle sortie est connectée aux entrées chargement des bascules d'états s 01B12 et 03B12, a pour effet de transférer dans la partie état de la mémoire le mot d'état précédemment chargé dans le registre 04A12.

Le caractère de donnée reçu est alors transféré de la mémoire à accès sélectif interne du microprocesseur dans le registre 04A12, puis dans la partie données de la mémoire premier-entré premier-sorti. Le transfert dans le registre 04A12 se fait sous la commande de l'instruction WRRFMBD. Lorsque cette instruction est exécutée, le caractère reçu assemblé est appliqué aux huit sorties de données des microprocesseurs 01C10 et 07C10, lesquelles sorties sont connectées à l'entrée données du registre 04A12. Au même moment, la sortie bit 5 du décodeur 08C11, qui est connectée à une entrée de la porte 10D11, passe d'abord à l'état logique zéro, puis de nouveau à l'état logique un, c'est-à-dire qu'une impulsion à front négatif se trouve engendrée. Cette impulsion appliquée à la porte 10D11 fait apparaître à sa sortie, qui est connectée à l'entrée d'horloge du registre 04A12, des changements d'état similaires à ceux qui chargent le caractère dans ce registre.

Le caractère de donnée reçu ainsi chargé dans le registre 04A12 est alors transféré dans les circuits mémoires 06B12 et 08B12 au moyen de l'instruction WRRFMD. L'exécution de cette instruction fait passer la sortie bit 3 du décodeur 08C11 à l'état logique zéro; celle-ci se trouve mise à l'état un par lequel la sortie Q passe à l'état

logique un. Lors de l'exécution de l'instruction suivante, l'entrée CK de cette bascule se trouve à l'état logique un, la sortie Q reprenant l'état logique zéro lors de l'application de l'impulsion d'horloge suivante. Cette séquence de transitions d'état de la sortie Q, séquence qui se trouve appliquée aux entrées chargement des circuits mémoires 06B12 et 08B12, occasionne le transfert du caractère de donnée précédemment stocké dans le registre 04A12 dans la partie données de la mémoire de réception.

Le restant de la colonne est alors traité sous la direction du programme de commande de réception, d'une manière similaire à ce qui vient d'être décrit. Plus particulièrement, chaque bit de données reçu de l'équipement DCE local est introduit dans le microprocesseur, appliqué au registre de contrôle d'erreur, combiné à d'autres bits et assemblé en caractères, traité comme il convient, puis chargé dans la mémoire avec le mot d'état. Les bits de données zéro insérés à des fins de transparence par le transmetteur distant sont éliminés sous la direction du programme.

La colonne est traitée sous la commande du mot de commande réception (fig. 15d) chargé par le processeur de communications 10 dans les registres 01A10 et 05A10, afin de déterminer la longueur de chaque caractère de donnée d'une manière similaire à ce qui se passe pendant l'opération de transmission, comme décrit précédemment (puisque les messages transmis et reçus ont le même format). Plus particulièrement, le bit RCFX (bit 0) à l'état un spécifie un format de zone de commande étendu, le bit RAFX (bit 1) à l'état un spécifie un format de zone d'adresse étendu, le bit RTCB (bit 2) indique la présence d'un multiplet de contrôle de texte, le bit RCLF (bit 7) à l'état un indique la présence d'une zone de contrôle logique, les bits RBS (bits 4, 5 et 6) indiquent la longueur de multiplets des multiplets de la zone texte, et le bit RFCSIE (bit 3) indique si le bit FCS reçu doit être ou non introduit à la fin du message, comme il sera décrit plus loin. Ces bits sont, pour ce qui concerne le microprocesseur, les signaux RUPID0 à RUPID7, et ils peuvent être retraduits pour être chargés dans les registres 01A10 et 05A10, si cela est souhaité. Ces bits correspondent également aux bits CPDAT0 à CPDAT7, au connecteur 01C01 (fig. 1). Les caractères de données reçus de moins de 8 bits sont justifiés ou cadrés dans la mémoire à accès sélectif du microprocesseur par le programme de commande de réception, avant d'être chargés dans la mémoire en vue de leur transfert au processeur de communications. Si la transmission se fait en mode flot de bits (décrit plus haut dans la partie opération de transmission), chaque caractère est poussé avant d'être chargé dans la mémoire. Ce processus se poursuit jusqu'à la fin de la colonne.

Selon le protocole de commande de transmission de données à haut niveau, la fin d'une colonne est indiquée par l'une des deux séquences binaires uniques dans le flot de données. Plus particulièrement, une succession de 7 bits un indique une colonne abandonnée, tandis que la séquence 0111110 indique la fin d'une colonne normale. Le programme commande l'examen de chaque bit de données, conjointement aux bits reçus antérieurement, en vue de trouver les séquences décrites ci-dessus. On notera que l'une quelconque de ces séquences peut se produire à un moment quelconque, et non seulement à une limite de caractère. Lorsqu'une fin de colonne se produit, le programme commande l'exécution d'une série d'opérations de terminaison et engendre un mot d'état contenant des bits non nuls contenant des informations relatives à la colonne qui vient d'être reçue, et il commande le chargement du multiplet d'état dans les mémoires d'état, comme il sera décrit plus loin.

Lorsqu'une séquence d'abandon est détectée, cet état se trouve signalé au processeur de communications 10 par l'envoi d'un bit spécial RA dans le mot d'état (fig. 15e). Le programme de commande de réception met également à l'état un le bit fin de colonne REOF et il stocke le multiplet d'état complété dans la mémoire à accès sélectif du microprocesseur. D'une manière similaire, en cas de surcharge pendant la réception de la colonne, le bit RO réservé pour l'indication de cet état se trouve mis à l'état un. Le programme de commande de réception complète cette opération en transférant le

mot d'état de réception dans la partie état de la mémoire (comme il sera décrit plus loin).

Si une séquence supplémentaire de 8 bits un est reçue, cela indique qu'il existe une liaison libre. La réception de 15 bits un contigus indique donc un tel état. Cet état est reporté au processeur de communications par l'intermédiaire d'une opération de transfert d'état similaire à celle qui a été décrite plus haut, pour laquelle le bit d'abandon RA est à l'état un, mais cette fois c'est le bit fin de colonne REOF qui est à l'état zéro. Il est entendu que, chaque fois qu'un multiplet d'état se trouve chargé dans la partie état de la mémoire, un caractère de donnée, même s'il s'agit d'un caractère de remplissage, est également fourni à la partie données de la mémoire afin de remplir complètement les seize positions binaires.

Les colonnes se terminent normalement par une séquence code complémentaire, comme décrit plus haut (fig. 15h). Les seize bits qui précèdent cette séquence code complémentaire sont les bits de contrôle d'erreur FCS, qui sont stockés dans la mémoire à accès sélectif du microprocesseur en vue d'une utilisation ultérieure possible, comme on le verra plus loin. Les bits qui précèdent immédiatement les bits FCS constituent le dernier caractère de donnée de la colonne. La détection de la séquence fin de colonne par la logique de commande résulte en l'initialisation d'une série d'opérations comprenant le traitement du dernier caractère de donnée, la vérification du résidu de contrôle d'erreur, la génération d'un mot d'état fin de colonne et, éventuellement, le transfert de la séquence FCS reçue vers le processeur de communications 10.

Le dernier multiplet de la colonne est traité par la logique de commande de la manière suivante. Le nombre de bits de ce caractère reçus lorsque la séquence code complémentaire a été détectée est d'abord compté, et ce nombre est comparé à la longueur de multiplet effective, afin de déterminer si le dernier caractère est celui d'un multiplet complet ou d'un multiplet partiel. Un emplacement de 8 bits de la mémoire à accès sélectif du microprocesseur est utilisé pour stocker temporairement le mot d'état final pour chaque colonne. La position binaire 7 de ce mot (bit RPLB à la fig. 15e) est mise à l'état un par la logique de commande, afin d'indiquer un dernier multiplet partiel, ou bien cette position binaire est laissée à l'état zéro pour indiquer un multiplet complet. Le programme de commande commande également la génération et le chargement d'une valeur unique dans les positions binaires RLBS du multiplet d'état, afin d'indiquer le nombre de bits dans le dernier caractère.

La logique de commande sous la direction du programme exécute alors les opérations suivantes, toutes ces opérations étant similaires à celles décrites précédemment. Si le dernier multiplet contient moins de 8 bits, il est justifié ou cadré à droite dans la mémoire à accès sélectif RAM du microprocesseur. Si la transmission se fait en mode flot de bits, le caractère est poussé. Enfin, le dernier multiplet se trouve chargé dans la mémoire premier-entré premier-sorti.

La logique de commande sous la direction du programme de réception complète alors le mot d'état fin de colonne et charge celui-ci dans la mémoire de la manière suivante. On notera que ce qui suit ne s'applique pas au cas où une colonne est abandonnée ou en cas de surcharge, cas qui ont été discutés plus haut. Dans ces cas, les bits d'état RA et RO restent à l'état zéro. Le programme commande la mise à l'état un du bit fin de colonne REOF. Le seul bit d'état restant, le bit d'erreur RFCSE, est mis à jour comme décrit plus loin.

Pour vérifier l'intégrité de la colonne, la séquence de contrôle d'erreur CRC, qui a été engendrée sur la base de tous les bits de la colonne entre les codes complémentaires d'ouverture et de fermeture (sauf pour les bits zéro insérés à des fins de transparence), doit être examinée. Le programme commande d'abord la lecture du résidu de contrôle d'erreur provenant du générateur 04C11 où a eu lieu le calcul. Cette opération est effectuée au moyen d'une série de seize instructions RFCSR. L'exécution de chaque instruction RFCSR décale un bit de la séquence CRC accumulée du générateur CRC dans le microprocesseur.

Plus particulièrement, la sortie du générateur CRC est appliquée à une entrée de la porte — ET inversion 04D11. La sortie de celle-ci est connectée à la position binaire la moins significative du registre situé logiquement au sein de la mémoire à accès sélectif RAM du microprocesseur de réception. Pendant l'exécution de cette instruction, la sortie bit 5 du décodeur 06A11, qui est connectée à l'inverseur 07B11, passe à l'état logique zéro. La sortie de l'inverseur 07B11, qui est connectée à la porte 04D11, passe à l'état logique un afin de transférer la donnée de la sortie du générateur CRC au microprocesseur, comme décrit plus haut. Lorsque l'instruction RFCSR est exécutée, les entrées de commande de décalage du microprocesseur (bits 6, 7 et 8) spécifient une opération de décalage à gauche pour introduire le bit. Pendant l'exécution de cette instruction, la sortie bit 5 du décodeur 06A11, qui est connectée à la porte 06B11, passe à l'état logique zéro. La sortie de la porte 06B11, qui se trouve connectée à la porte 06C11, passe à l'état logique un, de manière à permettre que l'impulsion d'horloge suivante soit appliquée à travers cette porte à l'entrée d'horloge du générateur CRC, afin d'en extraire le bit suivant. Comme la sortie bit 6 du décodeur 06A11, qui est connectée à l'entrée de validation du générateur CRC, est à l'état logique un, le générateur exécute une opération de génération par décalage au lieu d'une opération de mise à jour.

Le résidu de contrôle d'erreur CRC, ayant été introduit dans le microprocesseur, est alors examiné par la logique de commande de réception afin de déterminer si la colonne reçue est exempte d'erreurs. On notera que le résidu est introduit dans le microprocesseur et vérifié par tranches de 8 bits, plutôt que d'être vérifié dans son entier en une seule opération. Le résidu réel pour une colonne bonne est FOB8 hexadécimal. Mais, comme le résidu a été d'abord inversé par la porte 04D11, puis décalé dans le microprocesseur de droite à gauche au lieu d'être de gauche à droite, la valeur pour une colonne exempte d'erreurs, vue dans le microprocesseur, est E2F0 hexadécimal. Le bit indicateur d'erreur RFCSE dans le mot d'état décrit plus haut est mis à l'état un si cette valeur n'est pas détectée, et il est laissé à l'état zéro si cette valeur est détectée.

La génération du mot d'état fin de colonne étant ainsi achevée, la logique de commande transfère ce multiplet d'état dans les mémoires d'état 01B12 et 03B12, par un processus à deux étapes utilisant le registre tampon 04A12. L'instruction WRRFMB est d'abord exécutée afin de transférer le mot d'état de la mémoire à accès sélectif du microprocesseur dans le registre tampon 04A12. Plus particulièrement, lorsque cette instruction est exécutée, le mot d'état stocké à l'adresse 5 dans la mémoire à accès sélectif RAM (où il vient d'être assemblé) est appliqué aux lignes de données de sortie du microprocesseur, lesquelles lignes sont connectées aux entrées données du registre 04A12. La sortie bit 5 du décodeur 08C11, qui est connectée à la porte 10D11, passe à l'état logique zéro pendant l'exécution de cette instruction, puis elle reprend son état logique un à la fin de l'instruction. La sortie de la porte 10D11, qui est connectée au registre 04A12, change d'état d'une manière similaire afin de charger le mot d'état dans le registre. L'instruction WRRFMS est ensuite exécutée, comme décrit précédemment, pour transférer le mot d'état du registre 04A12 dans les mémoires 01B12 et 03B12.

Cette opération achève le fonctionnement de la logique de commande de réception dans un cas normal. Si l'utilisateur désire examiner la séquence FSC réelle reçue du transmetteur distant, le processeur de communications 10 met le bit RFCSIE (bit 4) à l'état un dans le registre 01A10. La logique de commande examine ce bit lorsqu'il commence à traiter la fin de colonne. Si ce bit est trouvé à l'état un, la vérification du résidu CRC dans le registre 04C11 n'est pas exécutée. Au lieu de cela, la séquence FCS de 16 bits, qui a été stockée dans la mémoire à accès sélectif RAM du microprocesseur, se trouve chargée dans la partie données de la mémoire, multiplets de 8 bits par multiplet, en deux opérations séparées.

Il est entendu que, dans ce qui précède, la disponibilité des registres d'entrée de la mémoire premier-entré premier-sorti est toujours vérifiée par l'instruction d'essai RFMIR avant toute tentative de charger la donnée et le mot d'état dans la mémoire. Si ces registres

ne sont pas disponibles au début d'une colonne ou pendant la réception d'une colonne, il se produit un état d'exécution incomplète, comme décrit plus haut. Toutefois, à la fin d'une colonne, c'est-à-dire après réception de la séquence code complémentaire de fermeture, la logique de commande attend, si cela est requis, jusqu'à ce que la mémoire devienne disponible, puis elle fournit le caractère de donnée final et le mot d'état fin de colonne à la mémoire. De cette manière, une fois reçue, une colonne ne sera pas surchargée et détruite par une colonne suivante quelconque.

Des données additionnelles reçues de l'équipement DCE local sont alors synchronisées sur la séquence code complémentaire, assemblées en caractères et traitées de la même manière que décrit plus haut. Toute colonne subséquente est ainsi traitée de la même manière que ce qui vient d'être décrit. Il est entendu que le code complémentaire de fermeture pour une colonne peut également servir de code complémentaire d'ouverture pour la colonne suivante.

Cela termine la description de la logique de commande de réception et du programme de commande correspondant. La mémoire premier-entré premier-sorti isole effectivement cette fonction de commande de la fonction d'interruption du processeur de communications 10. Les données et mots d'état chargés dans les registres d'entrée de la mémoire, comme il vient d'être décrit, sont acheminés vers la sortie de la mémoire, occasionnent la génération d'un signal d'interruption pour le processeur de communications 10 et sont ensuite extraits par le processeur pour être traités. Cette interaction avec le processeur de communications 10 est décrite ci-après.

La donnée reçue et le mot d'état associé, chargés dans le registre d'entrée de la mémoire, comme décrit plus haut, sont acheminés vers le registre de sortie de la mémoire. Le signal de sortie FULL de chaque mémoire 01B12, 03B12, 06B12 et 08B12 passe à l'état logique un afin d'indiquer que la sortie de cette tranche particulière de 4 bits du mot total est prête. Le signal sortie prête pour chacune des quatre mémoires est appliqué à une entrée de la porte — ET 11B12. La sortie de cette porte passe à l'état logique un chaque fois qu'un mot d'état de 16 bits complet est disponible pour être transféré au processeur de communications 10. Lorsque ce signal est à l'état logique un, la bascule 11C12 est mise à l'état un lors de l'application de l'impulsion d'horloge suivante. Lorsque la sortie Q de cette bascule, qui est connectée à une entrée de la porte 11D12, est à l'état logique un, elle fait passer la sortie de la porte 11D12 à l'état logique zéro qui se trouve ainsi appliquée à la broche 9 du connecteur 01B01 (fig. 1). Ce signal à l'état logique zéro sert de signal d'interruption pour le processeur de communications 10, afin d'indiquer à celui-ci que la donnée reçue et le mot d'état associé sont prêts à être transférés.

Comme on l'a vu plus haut, le changement d'état du signal appliqué à la broche 9 du connecteur 01B01, de la valeur logique un à la valeur logique zéro, indique au processeur de communications 10 qu'une donnée et le mot d'état associé sont disponibles dans la mémoire de réception. Le processeur répond à ce signal d'interruption en lisant d'abord le mot d'état réception, puis le caractère de donnée dans les sections appropriées de la mémoire, comme décrit ci-après. On notera tout d'abord que les huit sorties de données des mémoires d'état 01B12 et 03B12 sont connectées aux entrées données 1 des multiplexeurs 02C12 et 06C12, et que les huit sorties données des mémoires de données 06B12 et 08B12 sont connectées aux entrées données 0 de ces multiplexeurs. Lorsque l'entrée sélection (borne 1) des multiplexeurs est à l'état logique zéro, les entrées de données se trouvent sélectionnées par le multiplexeur et, lorsque l'entrée de sélection est à l'état logique un, ce sont les entrées d'état qui sont sélectionnées. Les huit sorties desdits multiplexeurs sont connectées aux huit nœuds de fonction ET câblés 01D04 à 08D04 (fig. 4), qui sont eux-mêmes connectés aux broches 1 à 6, 8 et 9 du connecteur 01C01, lesquelles broches se trouvent connectées au processeur de communications 10.

Les sorties des multiplexeurs 02C12 et 06C12 sont normalement à haute impédance mais, lorsque le processeur de communications 10 lit une donnée ou un mot d'état dans les mémoires de réception,

l'entrée F de ces multiplexeurs, c'est-à-dire l'entrée sortie validée, passe à l'état logique zéro qui permet à l'information présente aux sorties sélectionnées de la mémoire d'être transférée au processeur de communications 10. Celui-ci lit le mot d'état de réception en appliquant des signaux d'adresse et de contrôle au décodeur d'adresse réception 04A02. La sortie bit 0 est ainsi portée à l'état logique zéro, tandis que la sortie bit 6 reste à l'état logique un. Comme la sortie bit 0 est connectée à la porte 04D02, la sortie de celle-ci passe à l'état logique zéro et, comme cette sortie est connectée à l'entrée F des multiplexeurs 02C12 et 06C12, ces multiplexeurs se trouvent utilisés. Comme l'entrée sélection de ces multiplexeurs, qui est connectée à la sortie bit 6 du décodeur 04A02, est à l'état logique un, les sorties des mémoires d'état 01B12 et 03B12 se trouvent sélectionnées et l'information d'état est envoyée au processeur de communications 10.

Ce processeur lit ensuite le caractère de donnée reçu, d'une manière similaire à celle qui vient d'être décrite, sauf que les lignes d'adresse et de contrôle du décodeur font cette fois passer la sortie bit 6 à l'état logique zéro, tandis que la sortie bit 0 reste à l'état logique un. Comme la sortie bit 6 est également connectée à la porte 04D02, les multiplexeurs 02C12 et 06C12 se trouvent utilisés; mais, comme l'entrée sélection de ces multiplexeurs est cette fois l'état logique zéro, ce sont les sorties des mémoires 06B12 et 08B12 qui se trouvent sélectionnées et c'est la donnée reçue qui est envoyée au processeur de communications 10. La sortie bit 6 du décodeur 04A02 est également connectée à la porte 02B02. Lors de l'application du signal de pointage à la seconde entrée de cette porte, une impulsion à front positif se trouve engendrée à la sortie de ladite porte. Ce signal est appliqué aux entrées OUT des quatre mémoires 01B12, 03B12, 06B12 et 08B12, indiquant ainsi à ces mémoires que les 16 bits d'information ont été extraits des registres de sortie. Cela permet à la donnée suivante contenue dans les quatre mémoires d'être transférée dans les registres de sortie. On notera que la sortie de la porte — ET 11B12 est à l'état logique un chaque fois qu'une donnée est disponible à la sortie des mémoires premier-entré premier-sorti, ainsi qu'il ressort de ce qui précède.

Le processeur de communications 10 a ainsi répondu au premier signal d'interruption de réception en transférant la donnée et le mot d'état associé. Un mot d'état contenant uniquement des bits zéro indique que le caractère de donnée associé est un caractère valable, qu'une fin de colonne ne s'est pas produite et qu'il n'existe aucun état de surcharge ou d'abandon. Le processeur applique alors des signaux d'adresse et de contrôle au décodeur d'adresse 04A02, afin de porter la sortie bit 2 de celui-ci à l'état logique zéro. Le processeur exécute ensuite l'une des deux opérations décrites ci-après.

Le processeur peut d'abord porter à l'état logique zéro le signal présent à la broche 20 du connecteur 01B01. Notant que ce signal et la sortie bit 2 du décodeur 04A02 sont tous deux appliqués aux entrées de la porte 05B02, la sortie de celle-ci passe à l'état logique zéro et la sortie de la porte 13B12, qui est connectée à l'entrée de remise à l'état zéro de la bascule d'interruption 11C12, passe également à l'état logique zéro. La bascule 11C12 se trouve ainsi remise à l'état zéro, supprimant le signal d'interruption, au moins momentanément. Le processeur de communications 10 se trouve ainsi mis en état d'attente. Comme la sortie de la porte 11B12 est appliquée à l'entrée CJ de la bascule d'interruption 11C12, cette bascule se trouvera remise à l'état un lors de l'occurrence de l'impulsion d'horloge dès qu'une donnée est disponible à la sortie de la mémoire, comme décrit plus haut. Il en résulte la génération d'un nouveau signal d'interruption pour le processeur de communications 10, ce qui a pour effet de répéter l'opération de lecture qui vient d'être décrite.

Le processeur de communications peut également effectuer une opération de lecture d'état équipement DCE similaire à celle qui a été décrite précédemment pour l'opération de transmission. On notera que la sortie bit 2 du décodeur 04A02 est à l'état logique zéro, tandis que les autres sorties des décodeurs 01A02 et 04A02, en particulier la sortie bit 2 du décodeur 01A02, sont à l'état logique un. La sortie bit 2 du décodeur 04A02 est connectée à une entrée de la porte 03D02 dont la sortie à l'état logique zéro est appliquée à

une entrée de la porte 02D02. La sortie à l'état logique zéro de celle-ci est appliquée à l'entrée sortie validée du multiplexeur 04B04, ce qui a pour effet de valider la sortie de l'inverseur 05A04 connectée à la broche 5 du connecteur 01C01.

On notera que les deux entrées de la porte 01D02, qui sont alimentées par les décodeurs 01A02 et 04A02, sont à l'état logique un, de sorte que la sortie de cette porte 01D02, qui est connectée à l'entrée sélection du multiplexeur 04B04, se trouve à l'état logique un. Le signal a ses entrées 1 ainsi sélectionnées, ainsi que la sortie de l'inverseur 05A04. De plus, les deux entrées du sélecteur 03A04 sont telles que le signal sortie prête de la porte 11B12, qui est également connecté au sélecteur 03A04, est appliqué à l'inverseur 05A04 et, de là, au processeur de communications 10. Celui-ci teste ce bit afin de déterminer si un autre caractère et son mot d'état associé sont disponibles à la sortie de la mémoire. Dans l'affirmative, le processeur de communications 10 commande la répétition de l'opération de transfert décrite plus haut. Si une information n'est pas disponible, le processeur effectue l'opération décrite plus haut, qui remet la bascule d'interruption à l'état zéro, et il se met en état d'attente. L'adaptateur de ligne 14 applique un signal d'interruption au processeur de communications, lorsque la sortie de sa mémoire premier-entré premier-sorti a de nouveau une information disponible.

Le processeur de communications répète les opérations décrites plus haut. Chaque fois qu'un signal d'interruption est reçu, le processeur lit d'abord le mot d'état reçu, l'examine, puis lit le caractère de donnée. Plus particulièrement, le processeur examine le bit fin de colonne REOF (bit 3) du mot d'état de réception. Ce bit reste à l'état zéro jusqu'à ce que le dernier caractère de la colonne soit présenté. A ce moment, le processeur examine le mot d'état plus en détail, afin de déterminer si la colonne a été abandonnée ou si un état de surcharge s'est produit et, dans la négative, ce qui est le cas normal, s'il y a une erreur FCS, s'il y a eu un dernier multiplet partiel et combien de bits sont valables dans le dernier multiplet associé du message. Ces bits d'état sont bien entendu les mêmes bits que ceux qui ont été décrits plus haut, au cours de la description de la logique de commande de réception.

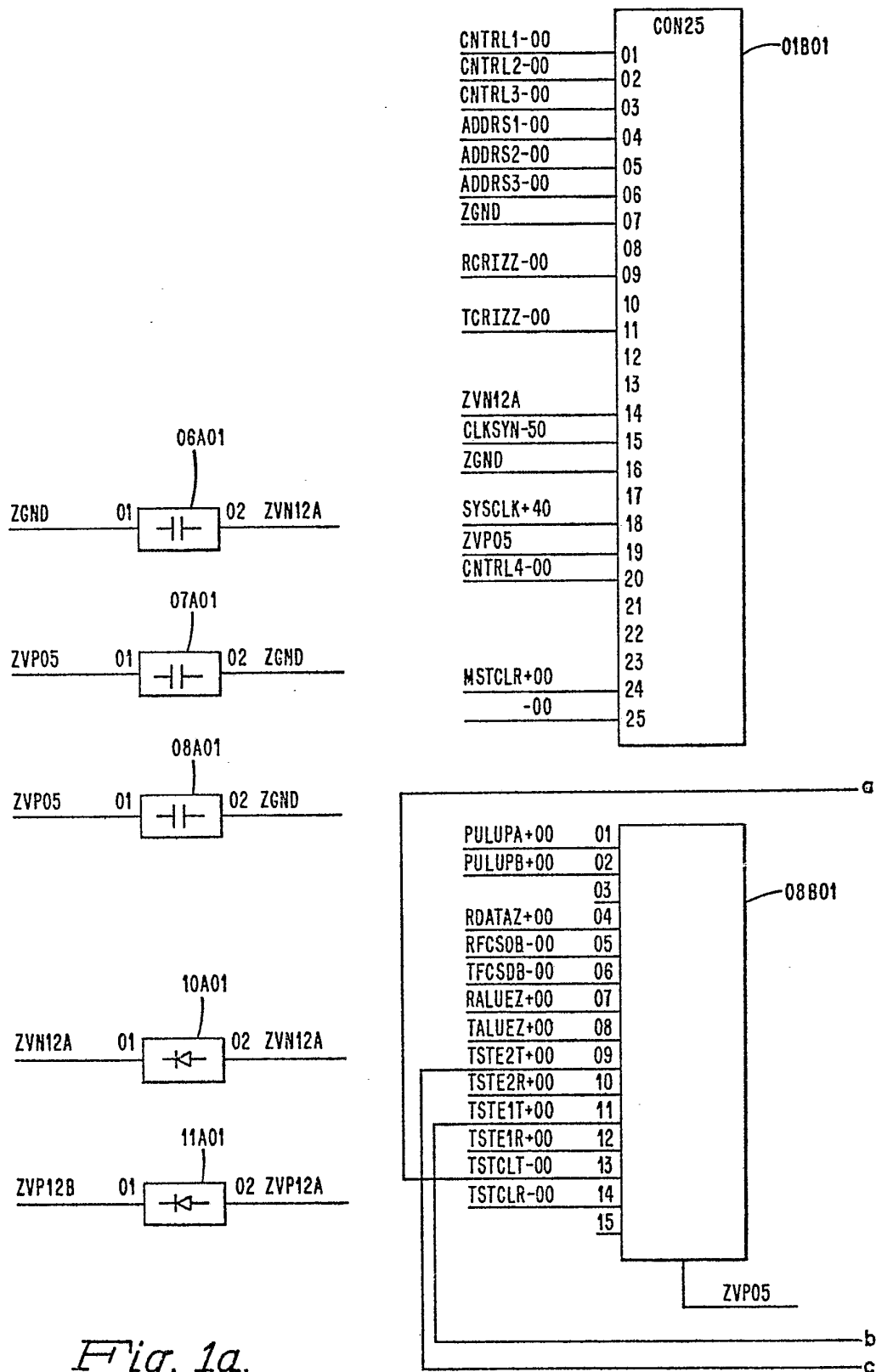
Après que la réception du dernier signal d'interruption pour la colonne en cours a été confirmée, un autre signal d'interruption peut être engendré. Lorsque le processeur lit le mot d'état et l'examine, la présence de bits zéro uniquement dans ce mot indique le début d'une nouvelle colonne ou, éventuellement, le bit d'abandon ou le bit de surcharge peut être mis à l'état un afin d'indiquer une liaison libre ou un état de surcharge. Lorsque des colonnes additionnelles sont reçues et traitées par la logique de commande de réception, elles sont chargées dans la mémoire comme décrit plus haut et engendrent des

signaux d'interruption pour le processeur de communications. La présence de données est indiquée par un mot d'état contenant uniquement des bits zéro. Le processeur de communications continue à répondre aux signaux d'interruption en transférant les informations d'état et les données à mesure qu'elles sont reçues. Le processeur peut commander à la logique de commande de réception d'écarter toutes les informations d'état et les données dans la mémoire, à un moment quelconque, en exécutant le processus de resynchronisation de réception décrit précédemment. Le processeur peut également inhiber les signaux d'interruption engendrés par la mémoire de réception en bloquant le récepteur, c'est-à-dire en chargeant un bit zéro dans la position binaire 6 du registre 08A04/11A04, comme décrit précédemment. Ce bit zéro, qui est appliqué à la porte 11D12, inhibe toute réception ultérieure d'un signal d'interruption en maintenant la sortie de cette porte à l'état logique un. Les signaux d'interruption de transmission peuvent être inhibés d'une manière similaire. A cet effet, si le processeur de communications 10 bloque le transmetteur en chargeant un bit zéro dans la position binaire 6 du registre 08A04/11A04, lequel bit se trouve appliqué à une entrée de la porte 12D06, la sortie de cette porte, qui produit le signal d'interruption de transmission pour le processeur de communications 10, est maintenue à l'état logique un, ce qui a pour effet d'empêcher la génération d'un signal d'interruption.

L'adaptateur de ligne de communications mixte matériel/programmation câblée décrit permet d'assurer le transfert de messages dans le format d'une commande de transmission de données à haut niveau organisée par bit et qui procure une capacité de traitement de données accrue, afin de réduire la charge de traitement du processeur de communications principal, de manière à augmenter la capacité de trafic du système.

En outre, cet adaptateur est capable de surveiller la zone adresse, la zone commande de liaison, le multiplet de commande de texte, la zone de commande logique, la zone texte et la zone séquence de contrôle de colonne dans les messages transférés, de surveiller, de commander et de commander des messages de longueurs de multiplets variables, d'engendrer des bits de parité cyclique pendant les opérations de transmission et de réception, et de délivrer un caractère de parité approprié avec les messages qu'il manipule.

L'adaptateur reçoit donc du processeur de communications des mots de commande de transmission et de réception au début du transfert de données et exécute des traitements appropriés sur le flot de données sous la direction de la programmation câblée, afin d'effectuer divers contrôles parité cyclique, longueur de multiplet, format de zone étendu et variable, dernier multiplet et transfert de bloc sur les données transmises et reçues.



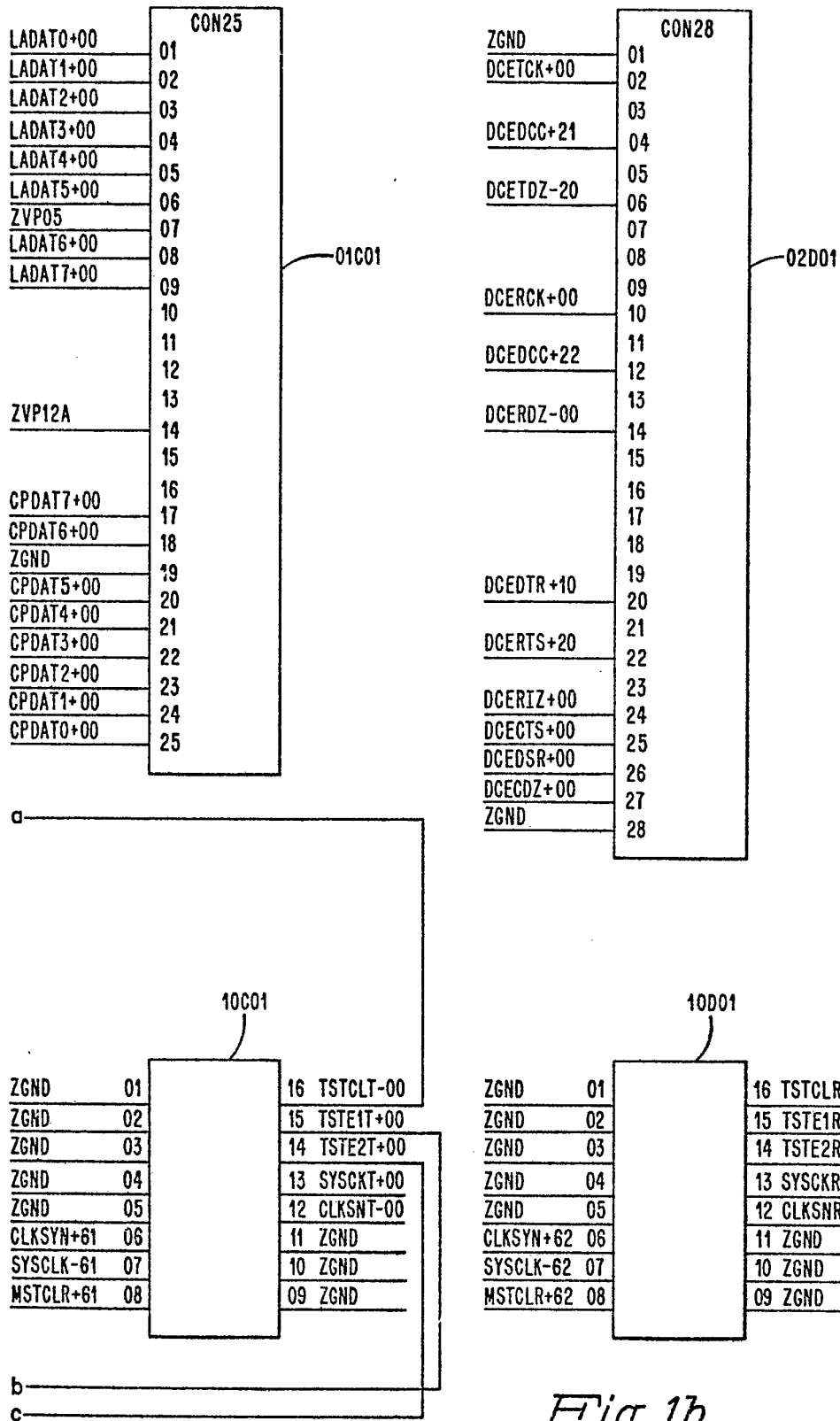
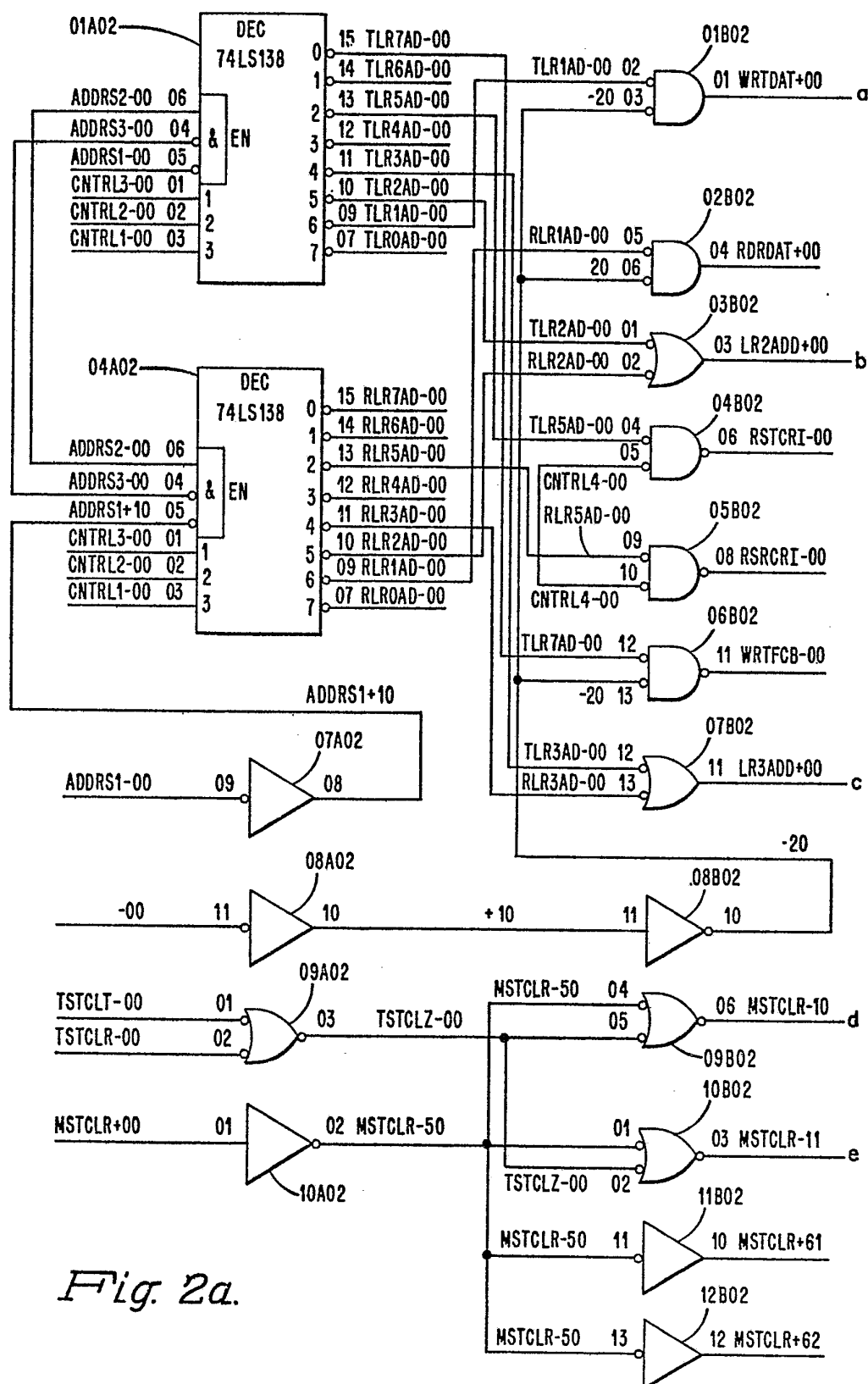


Fig. 1b.



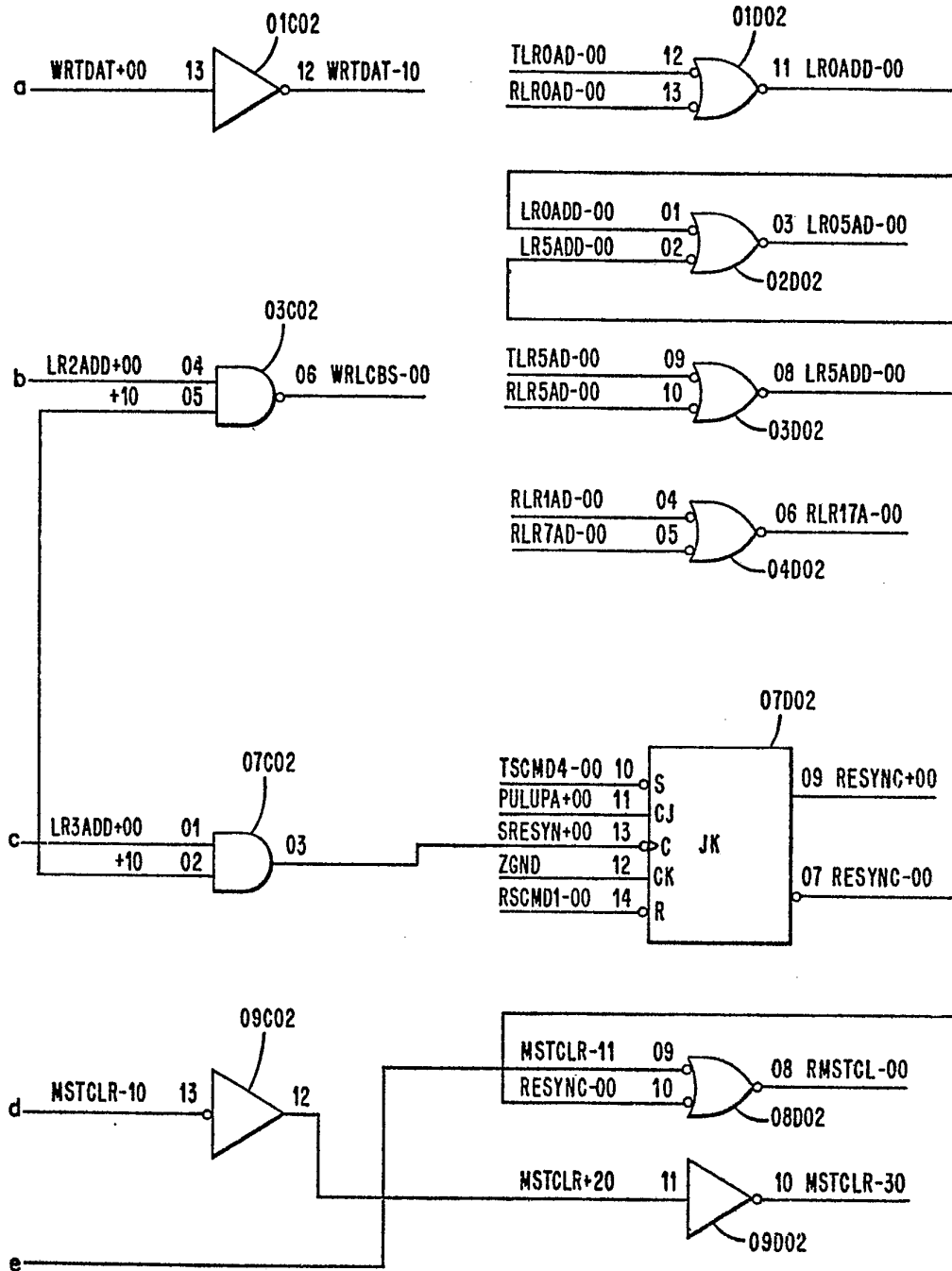


Fig. 2b.

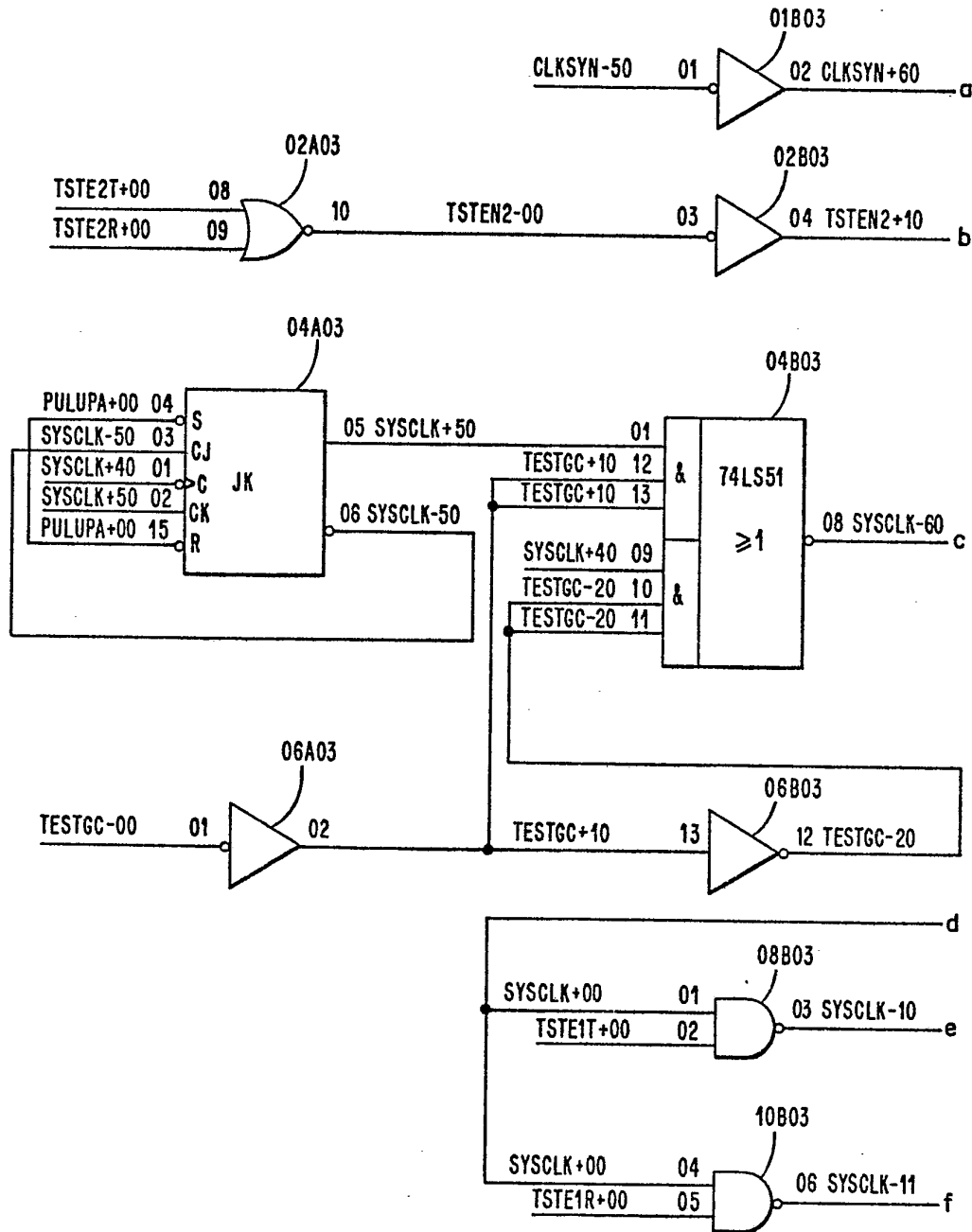


Fig. 3a.

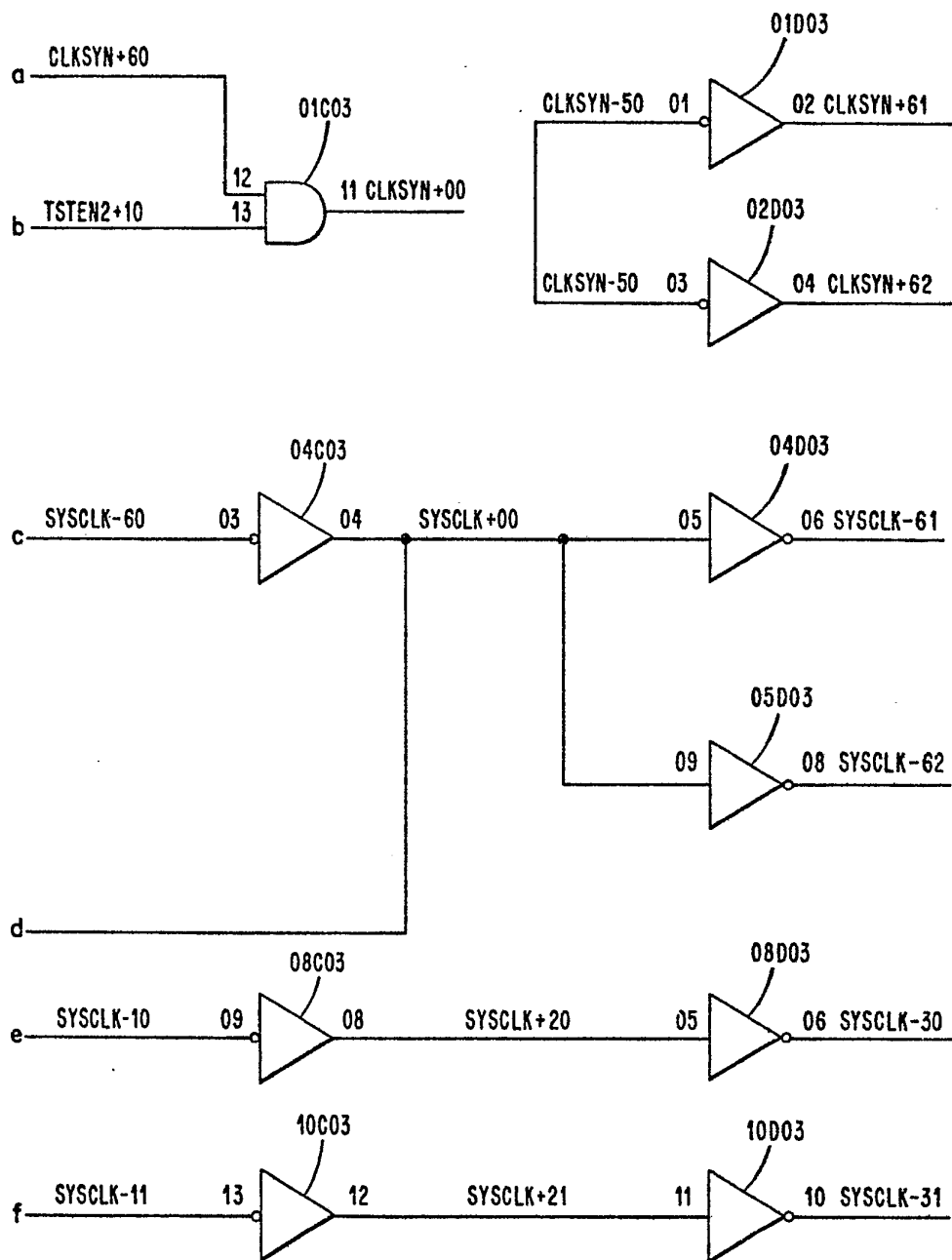


Fig. 3b.

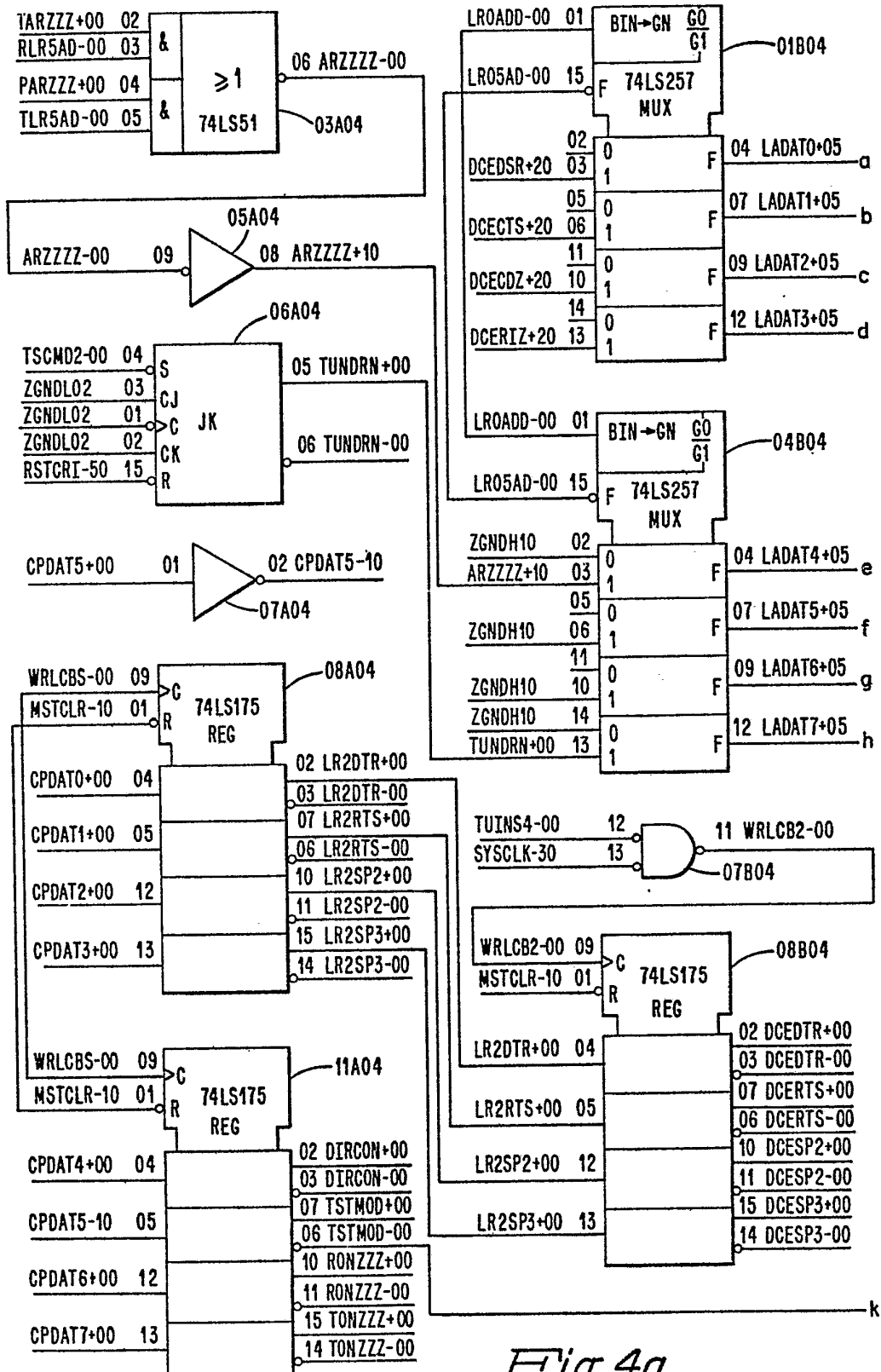


Fig. 4a.

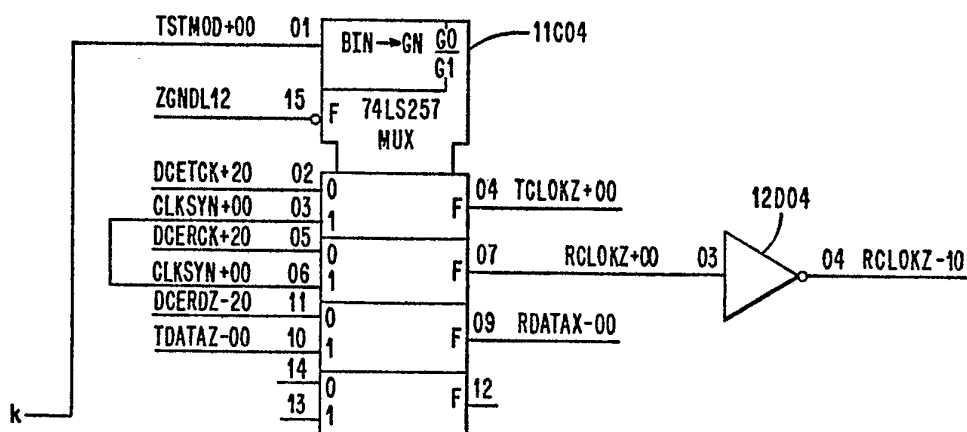
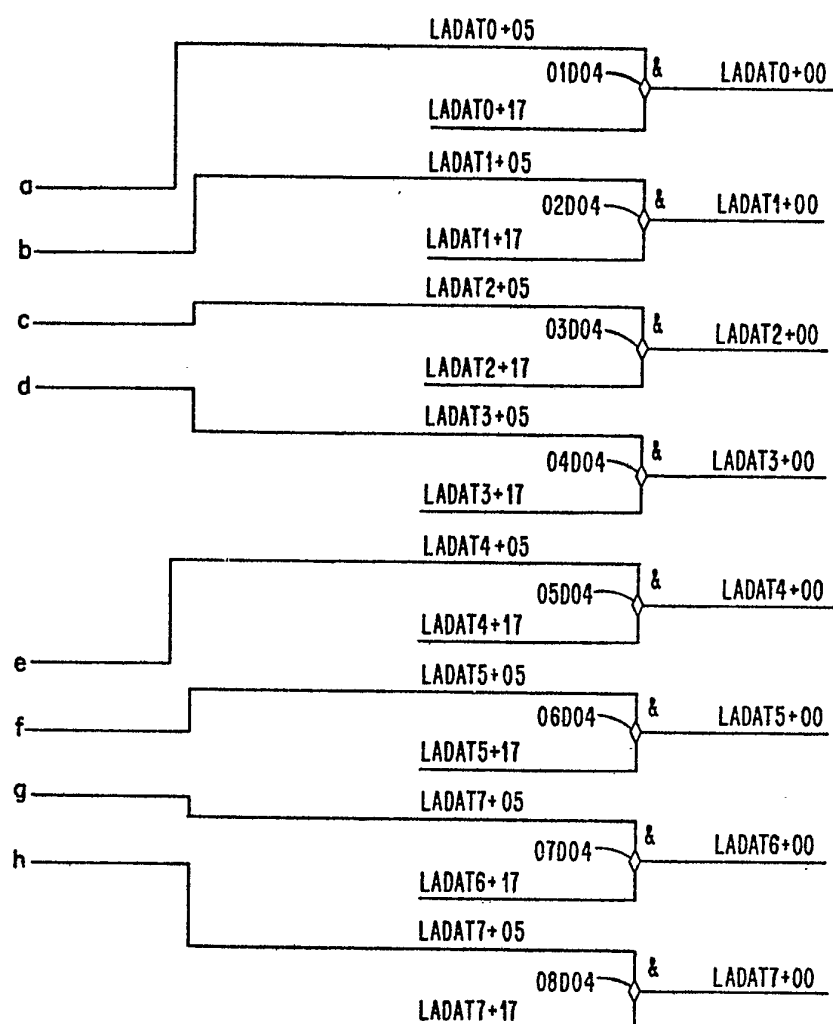
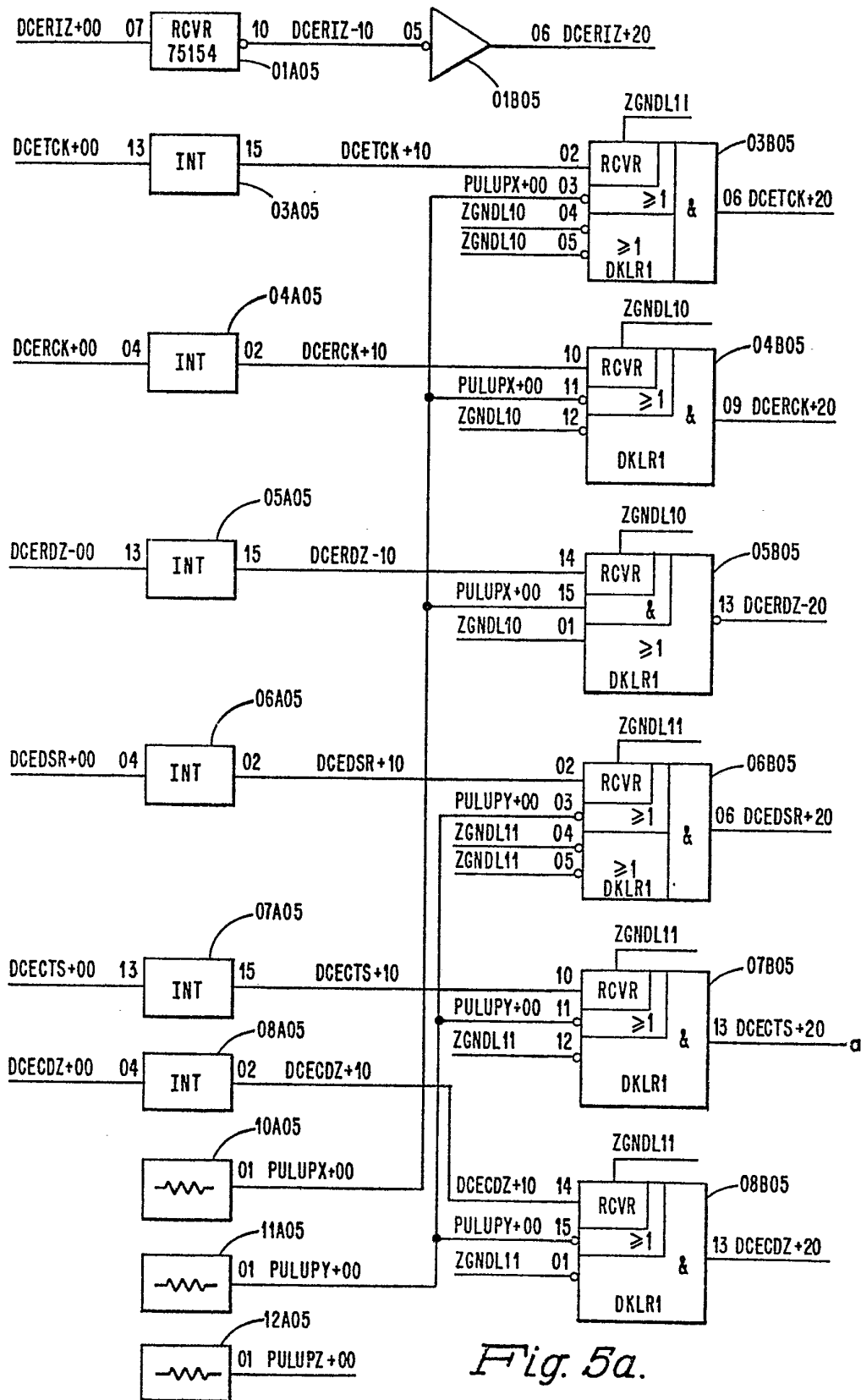


Fig. 4b.



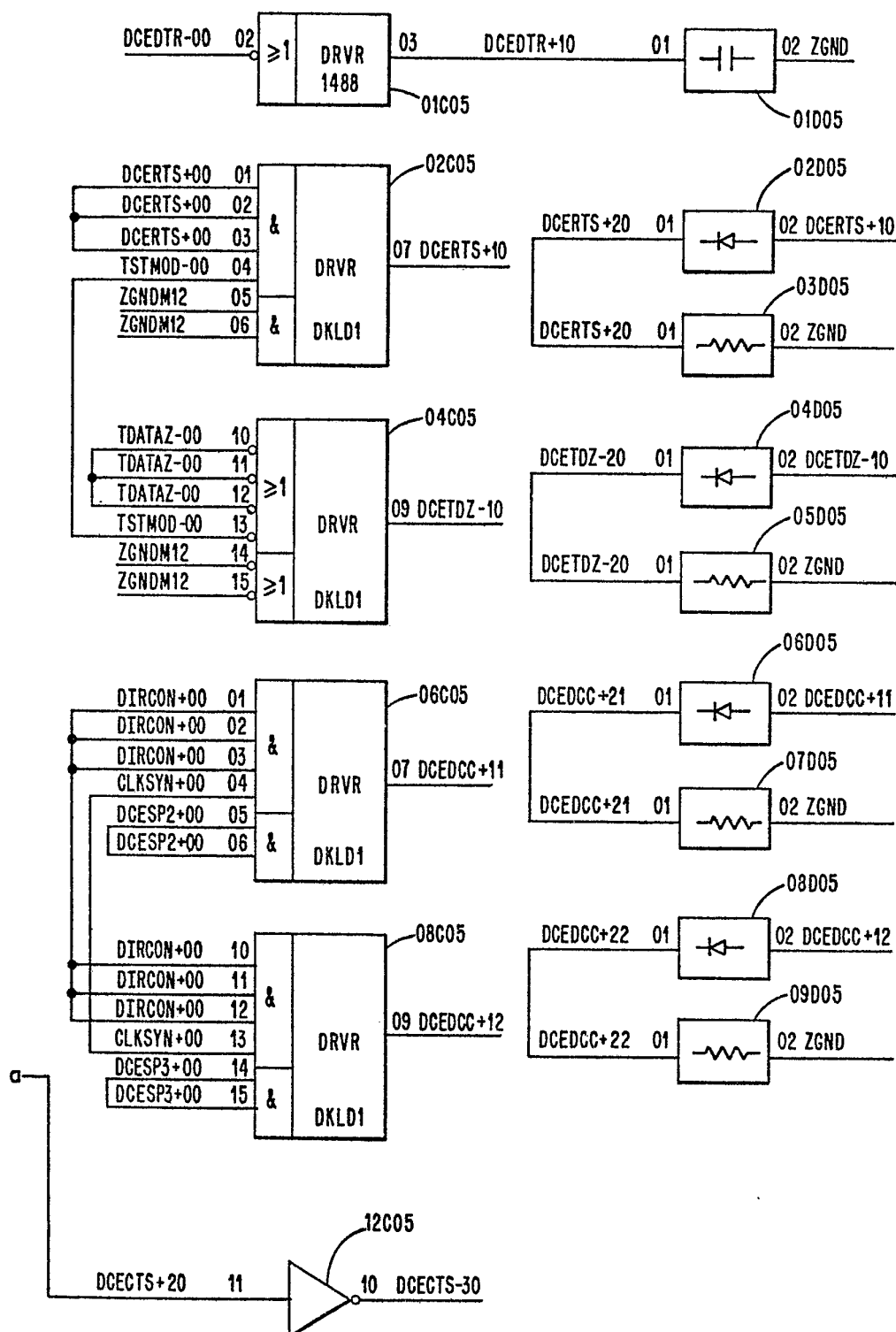


Fig. 5b.



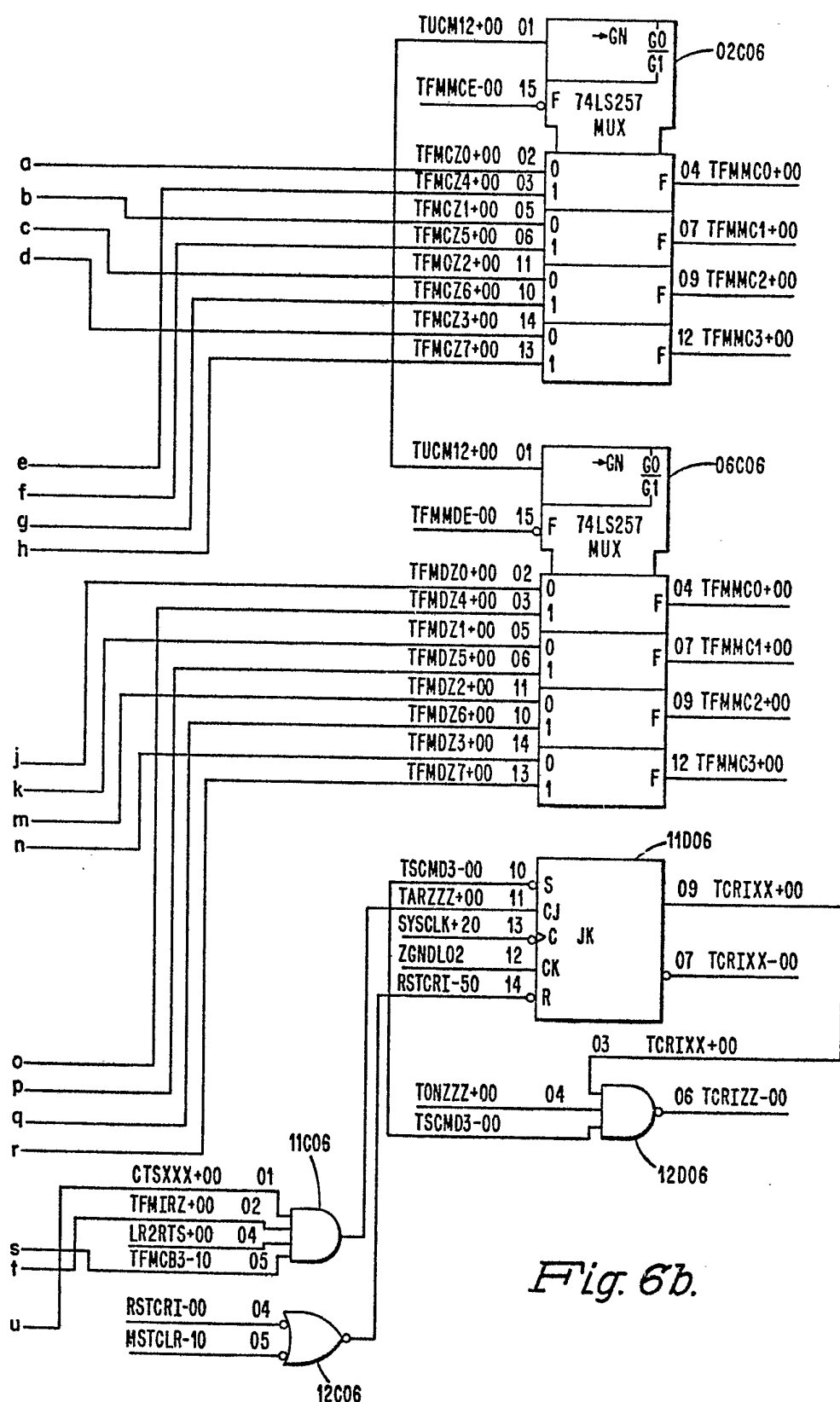


Fig. 6b.

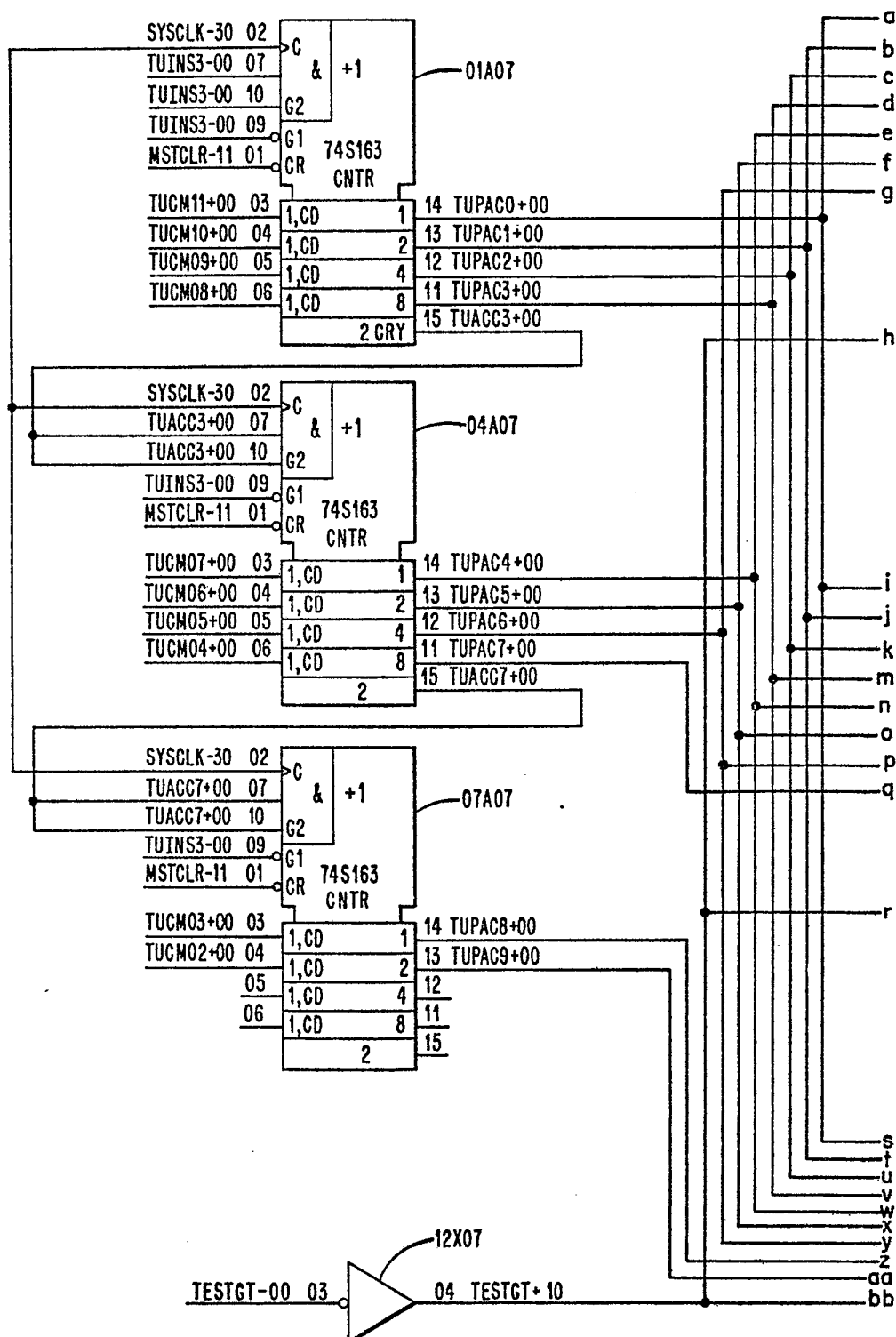


Fig. 7a.

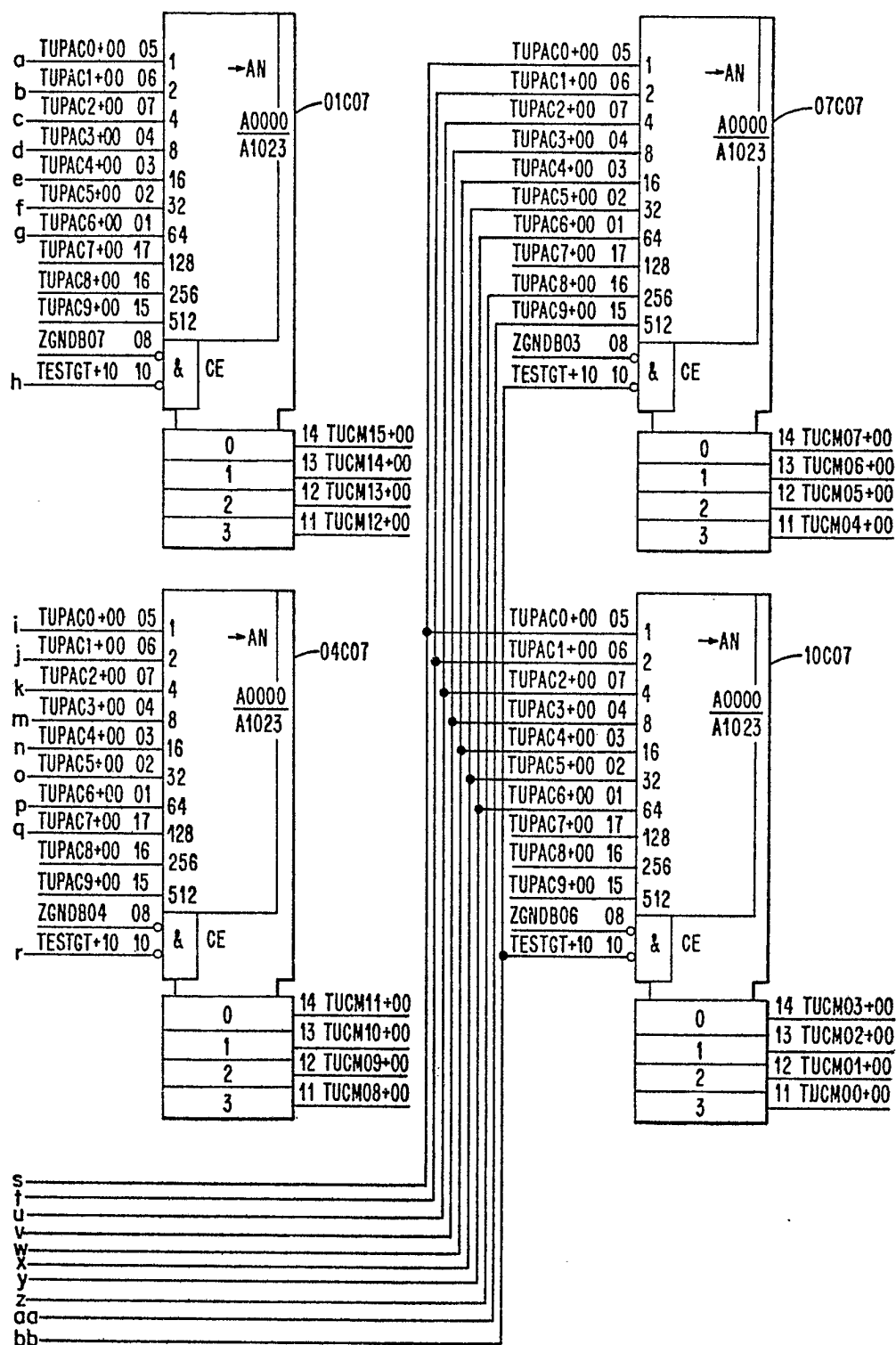


Fig. 7b-1

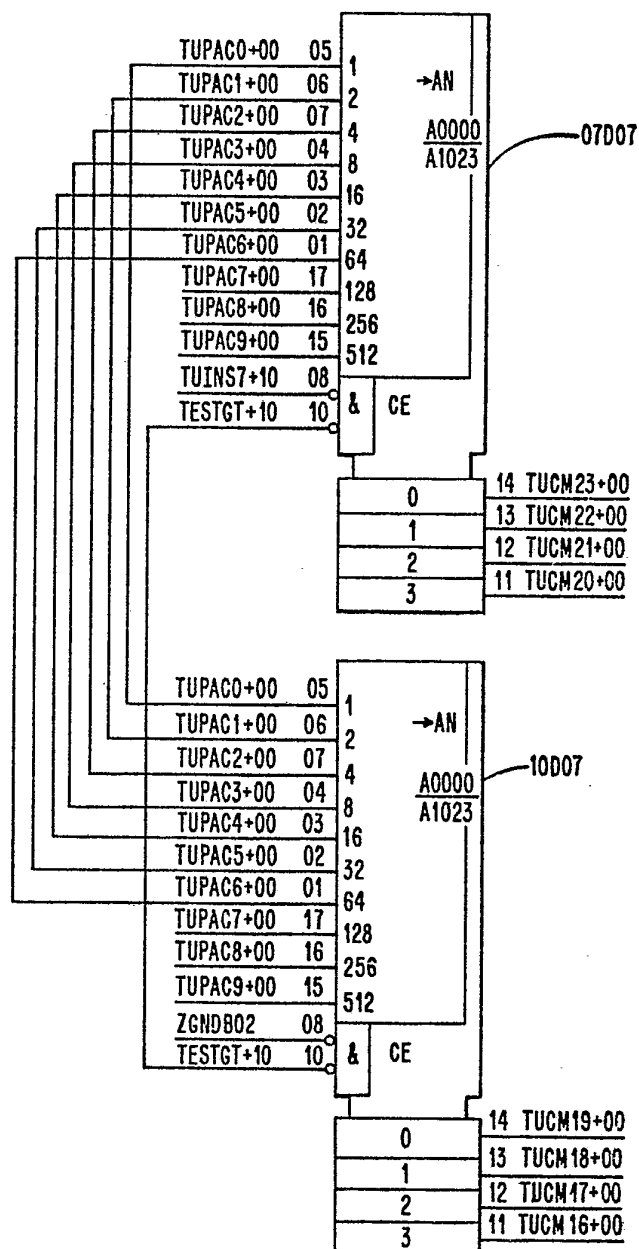


Fig. 7b.-2

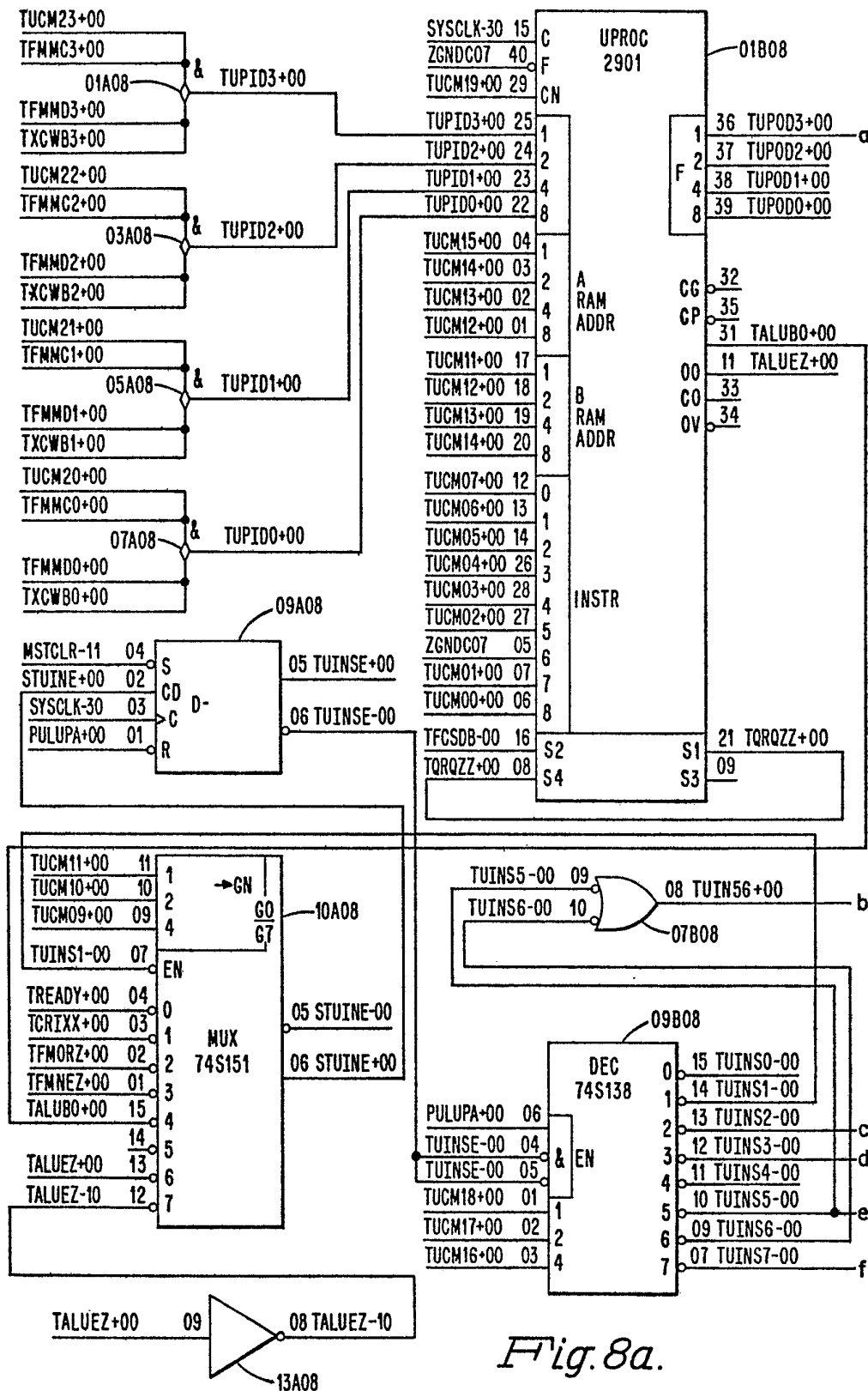


Fig. 8a.

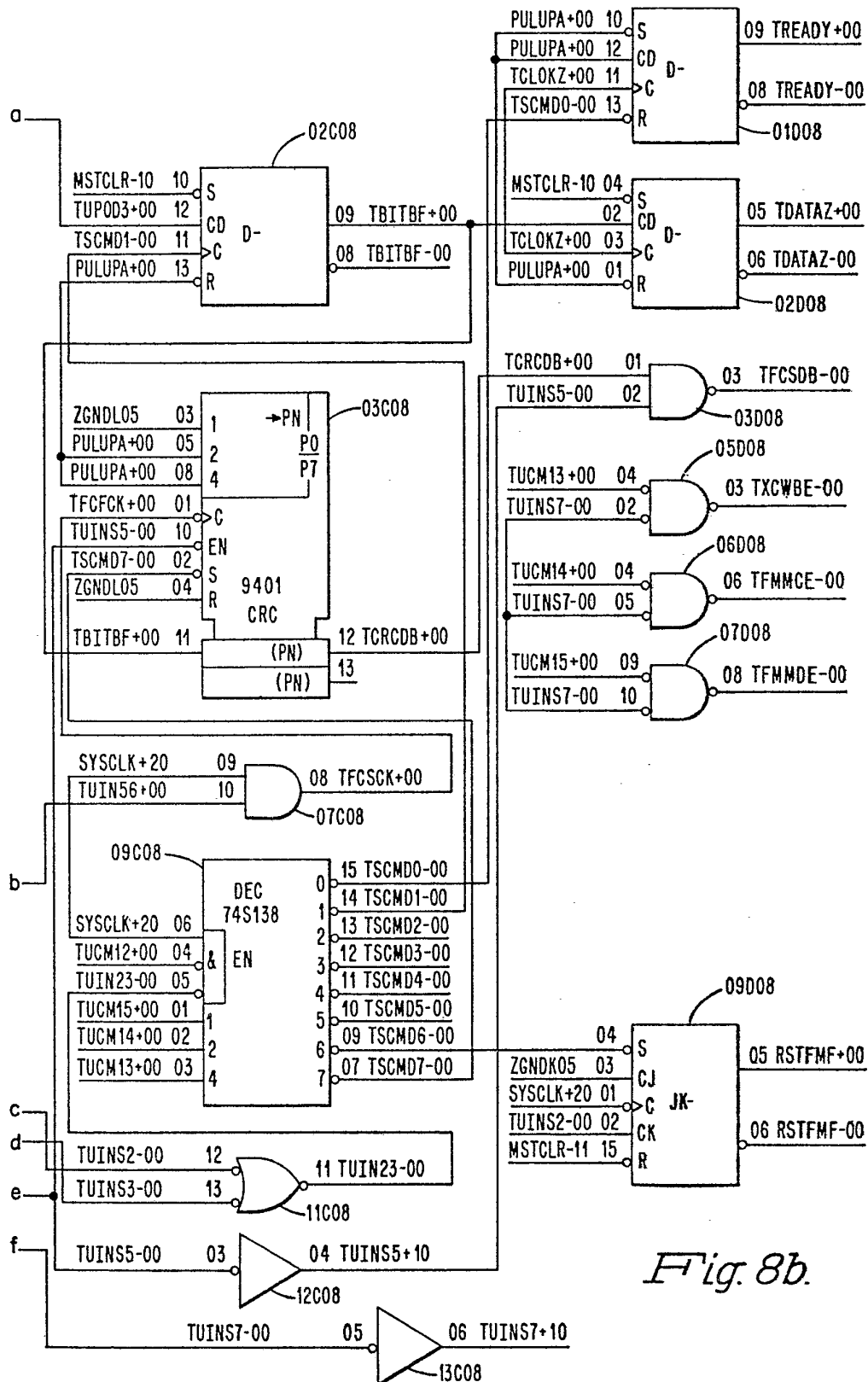


Fig. 8b.

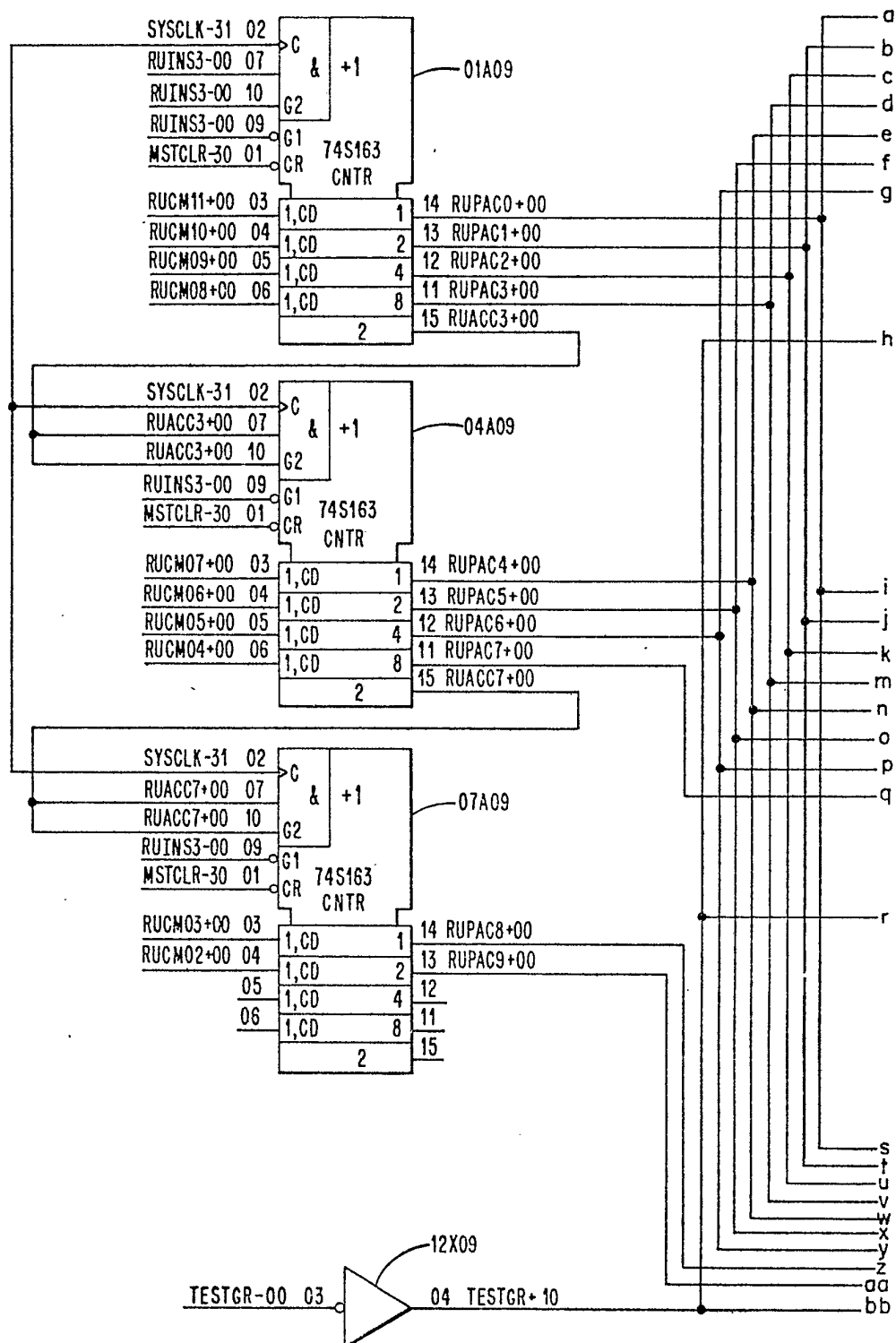


Fig. 9a.

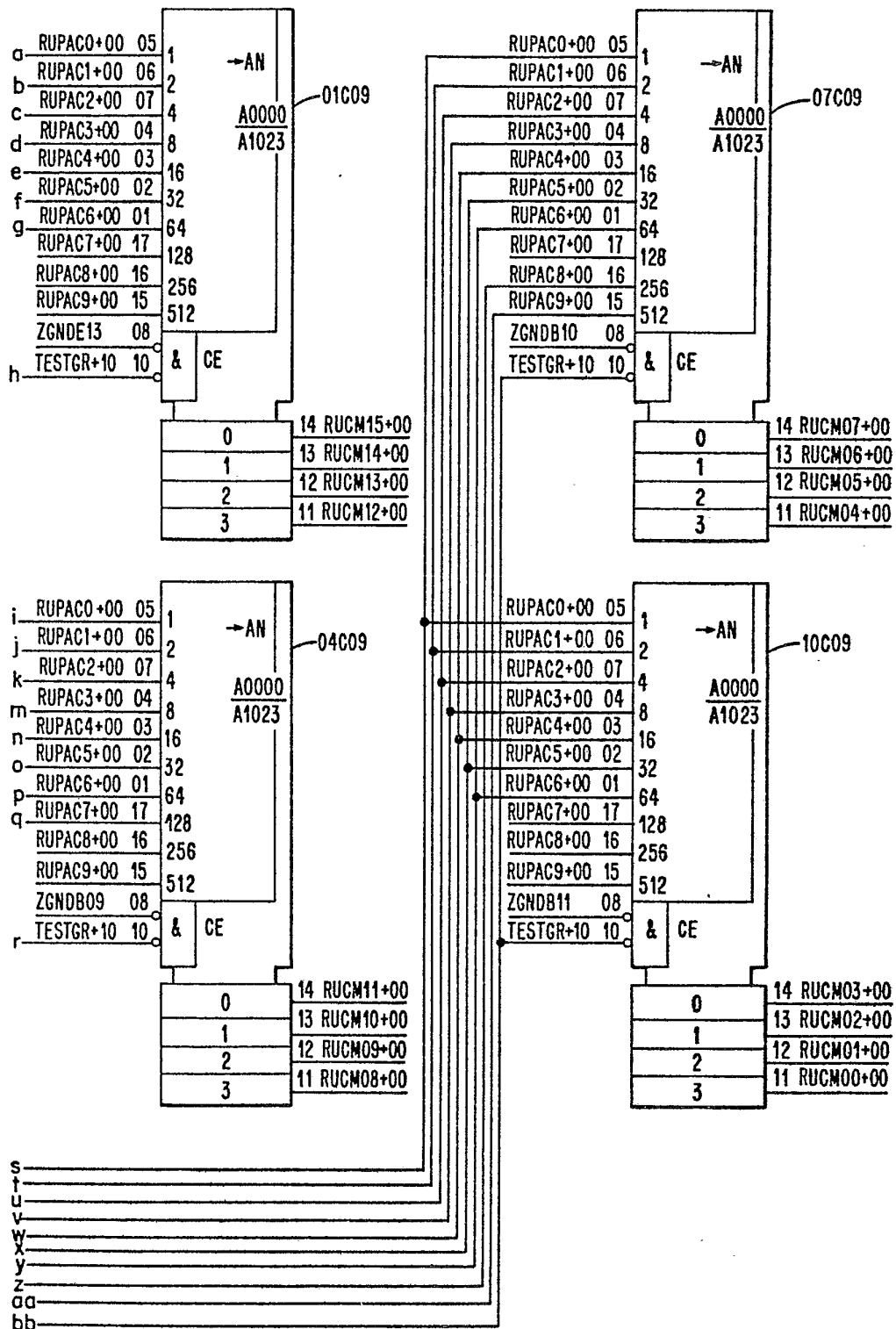


Fig. 9b-1

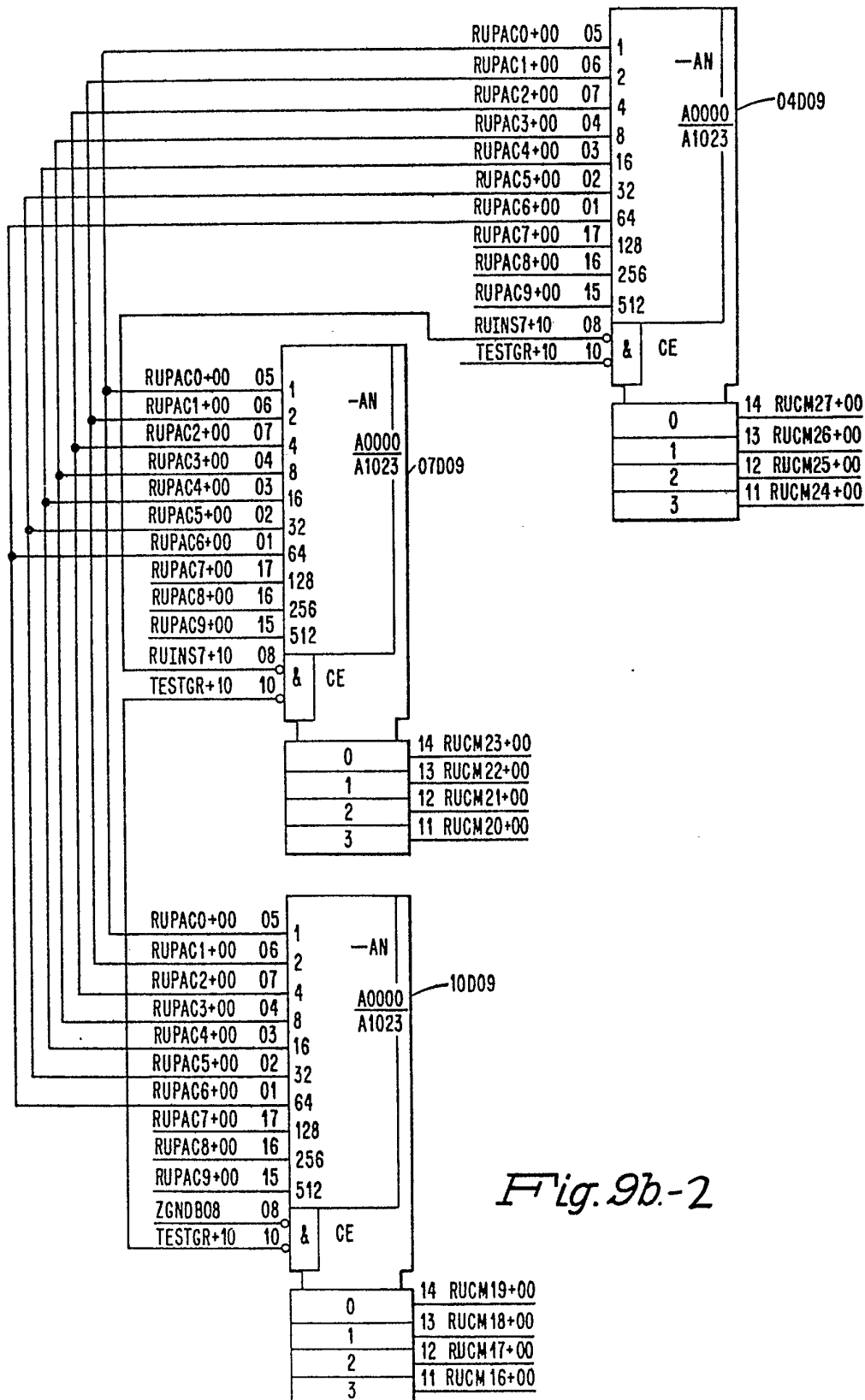


Fig. 9b-2

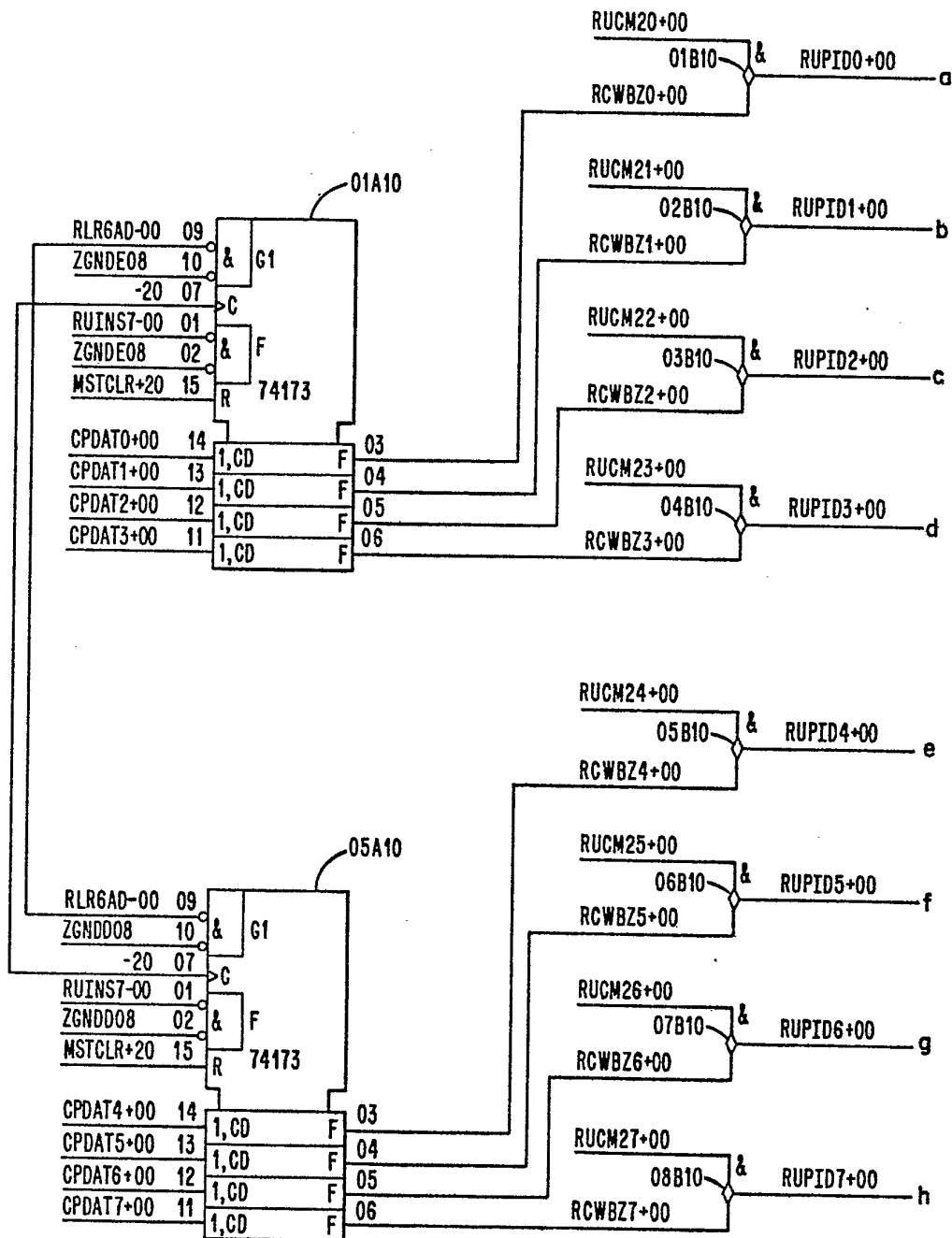


Fig. 10a.

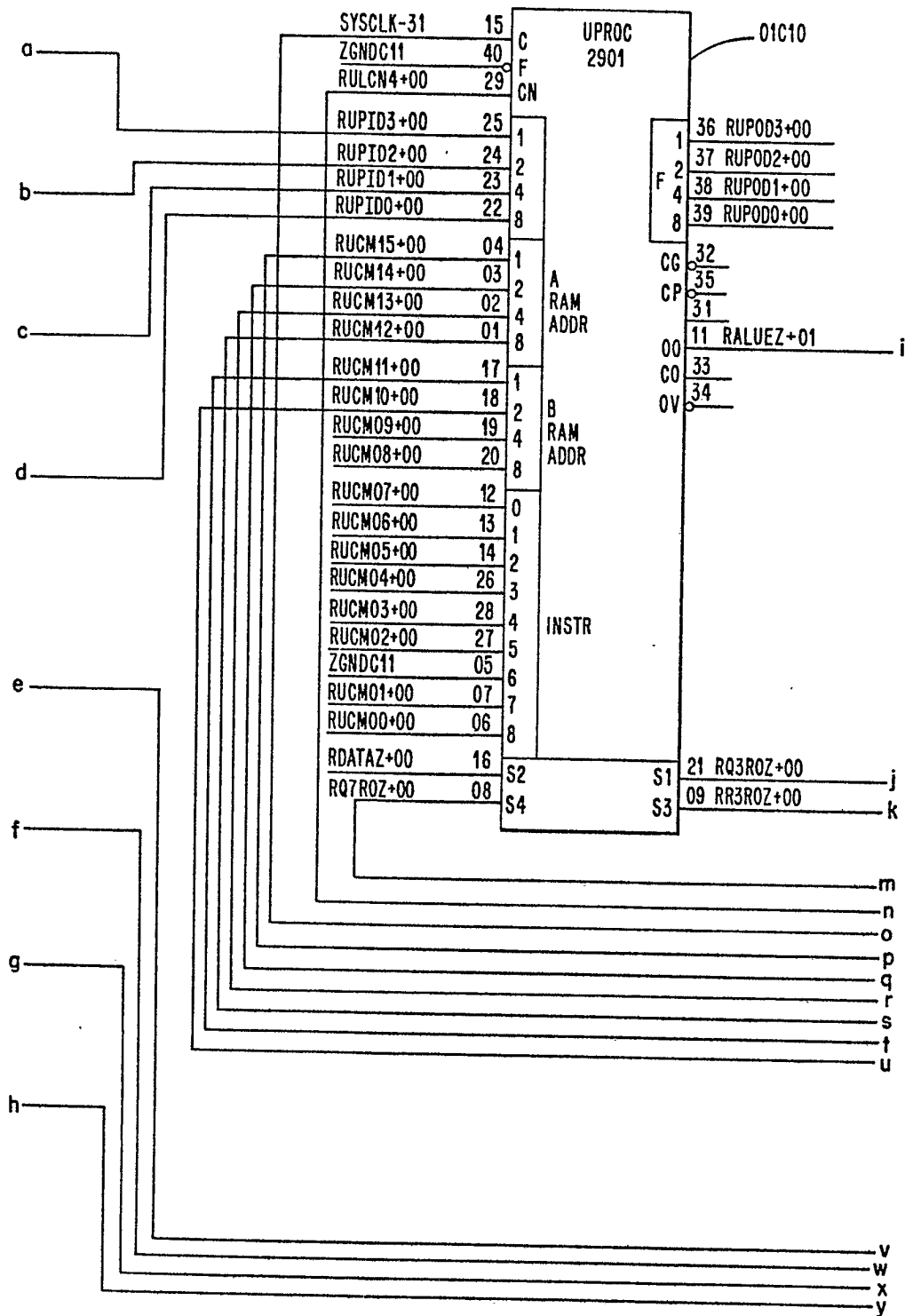
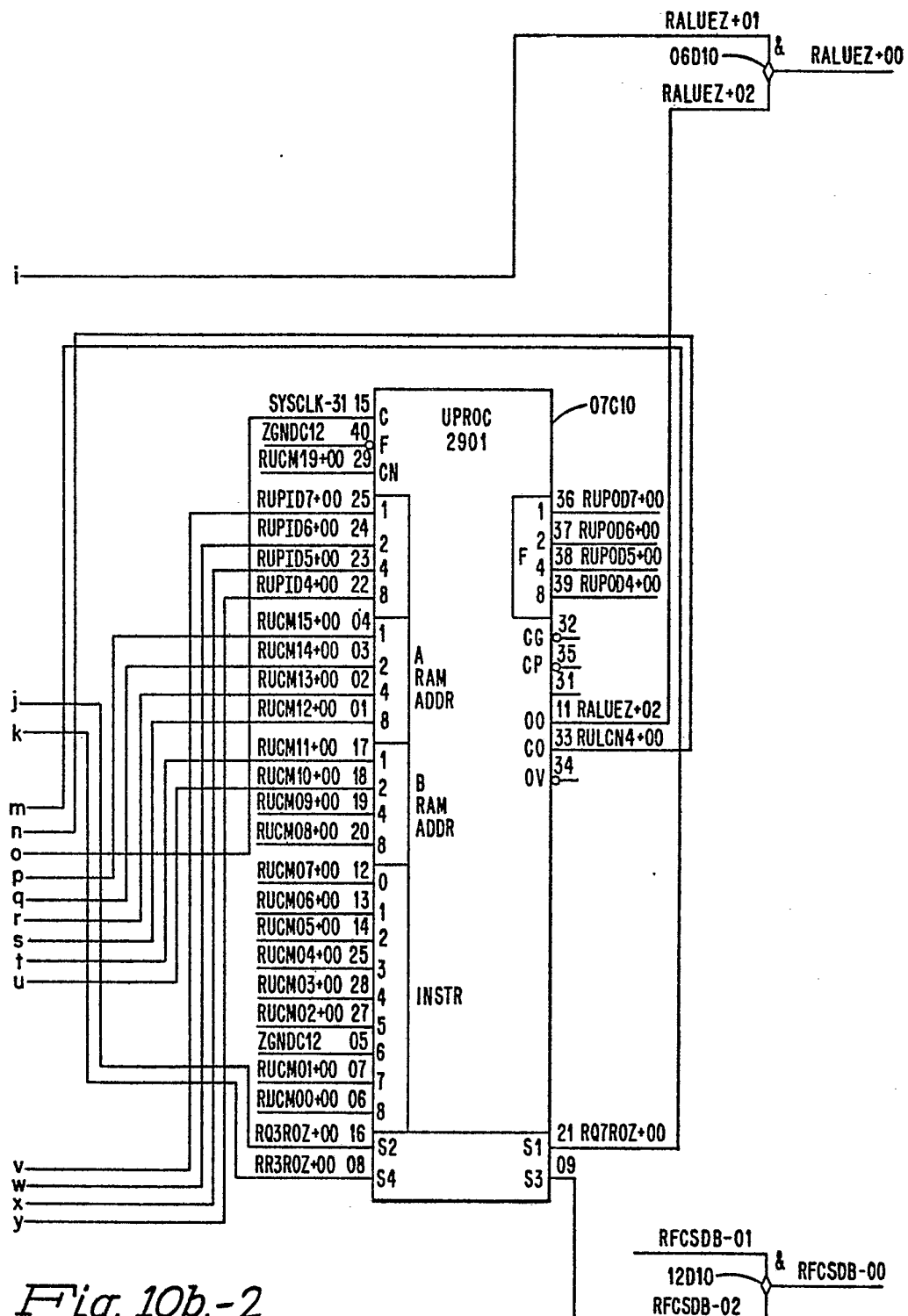


Fig. 10b.-1



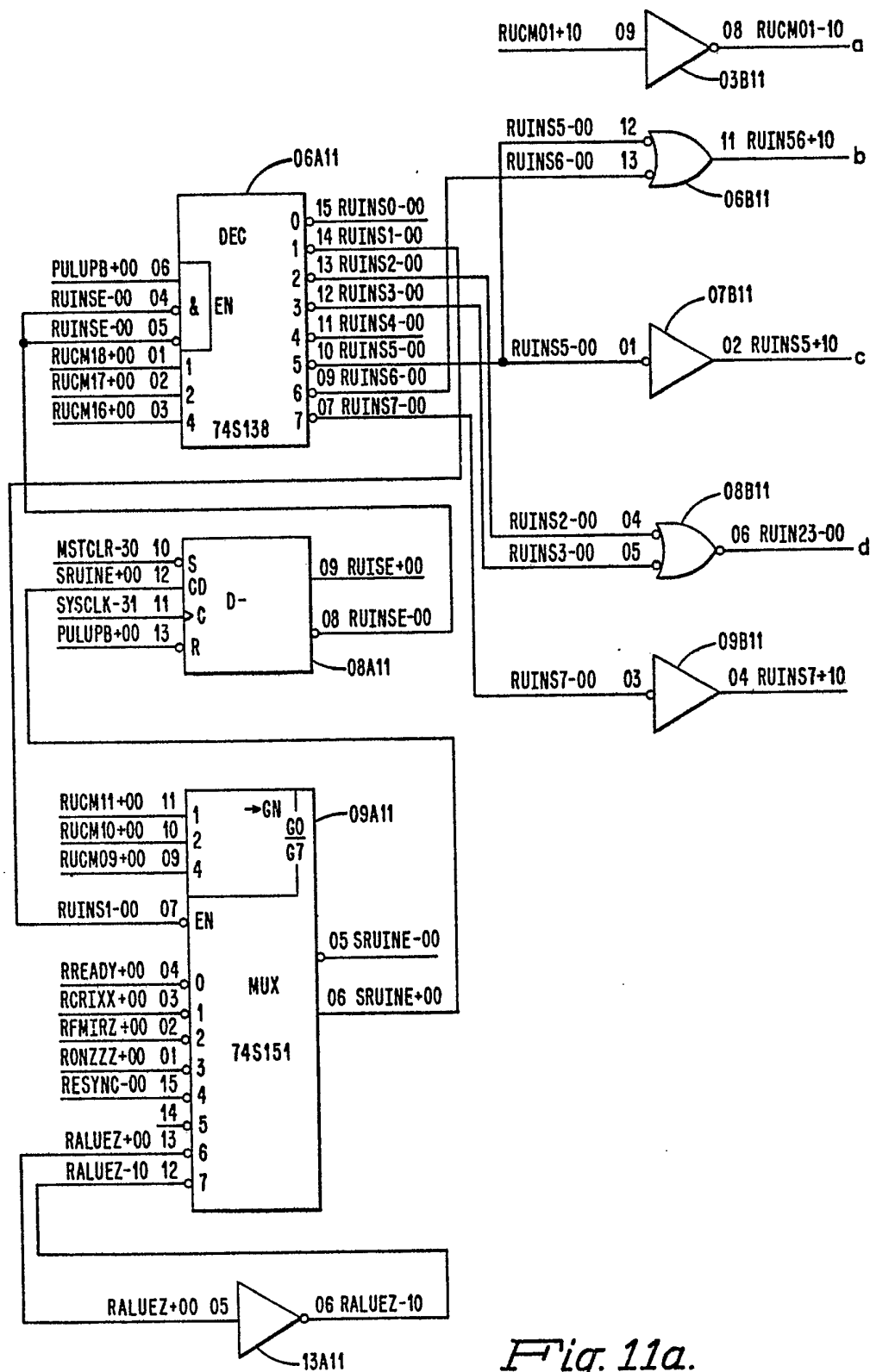


Fig. 11a.

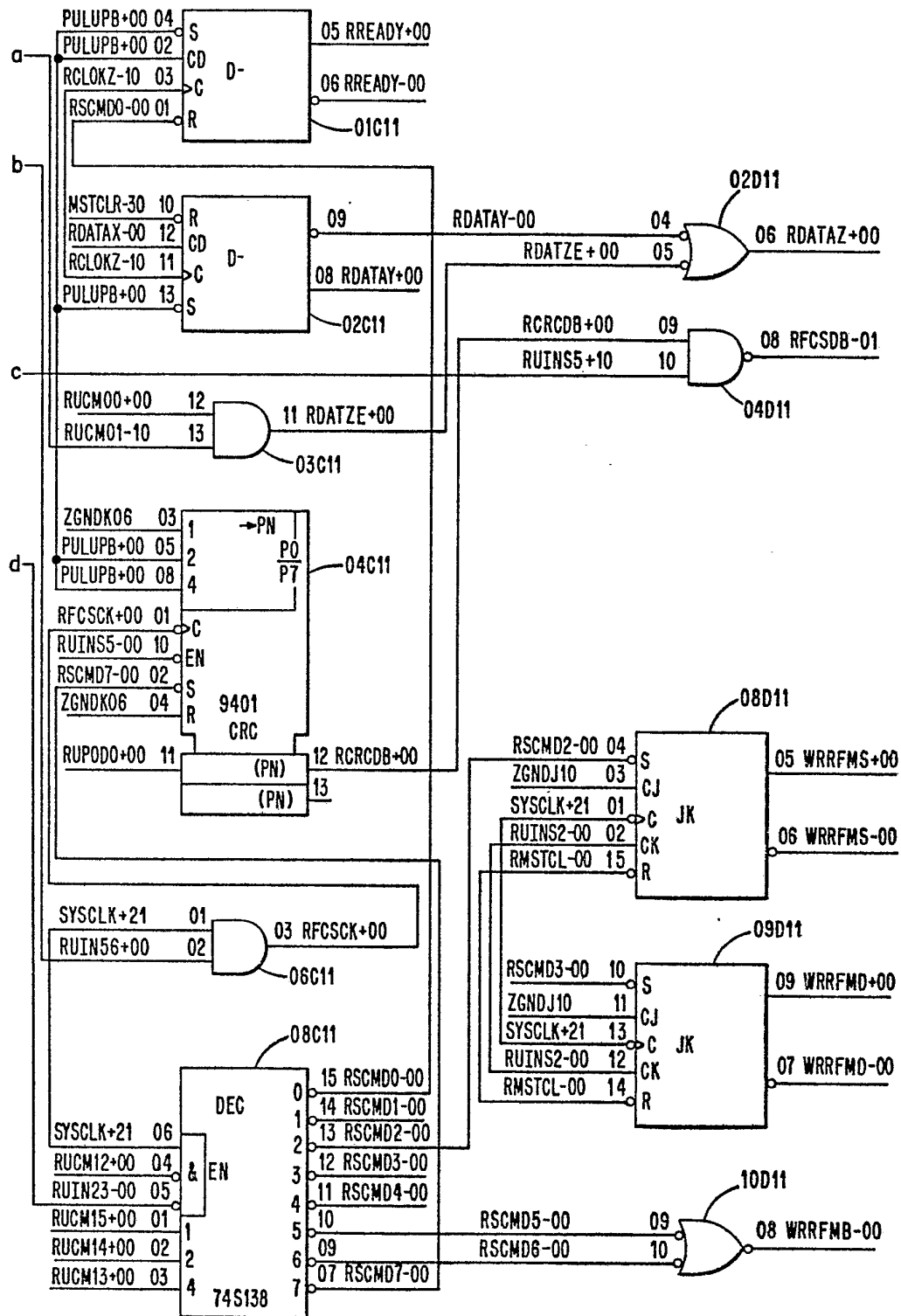


Fig. 11b.

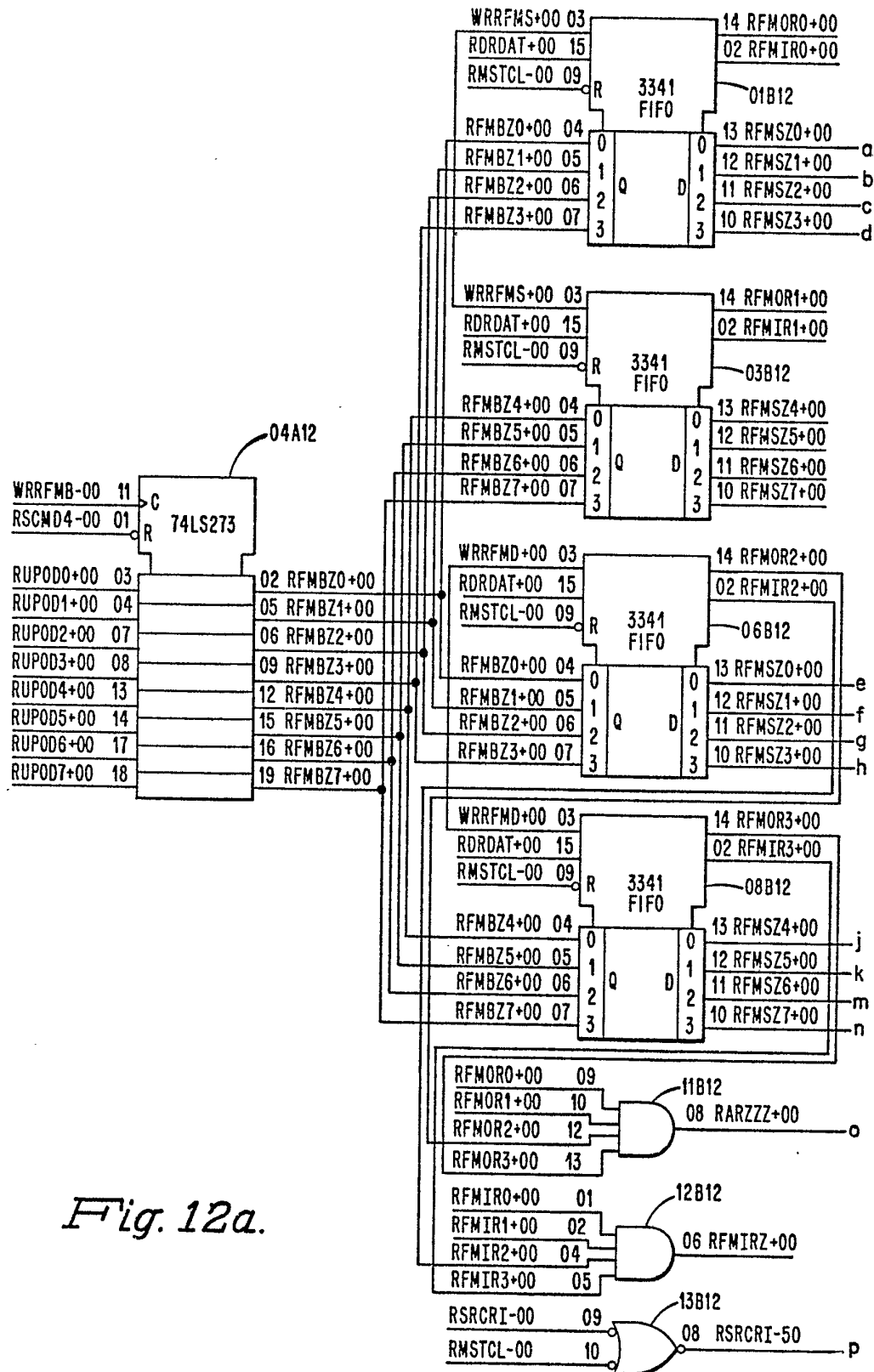
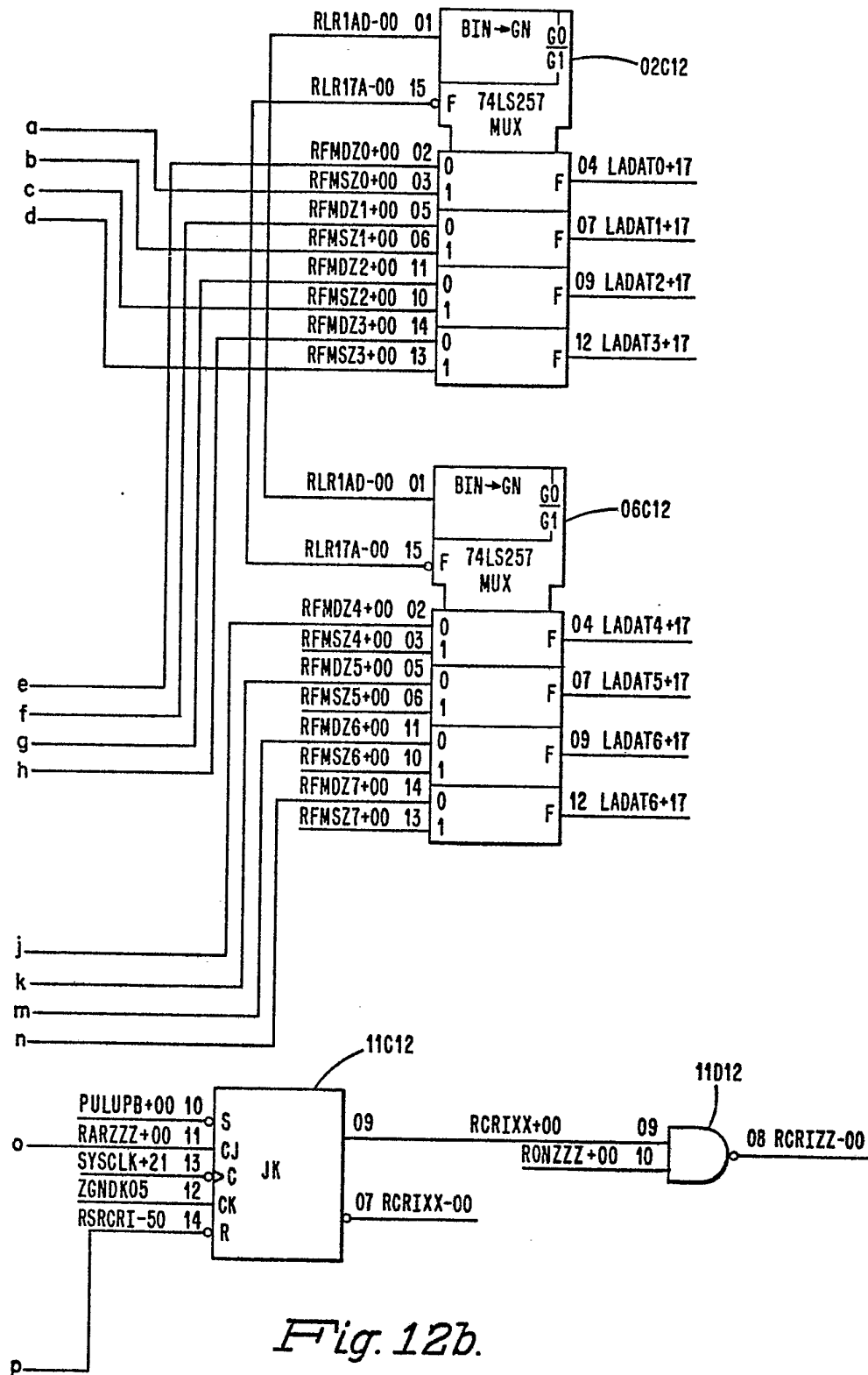
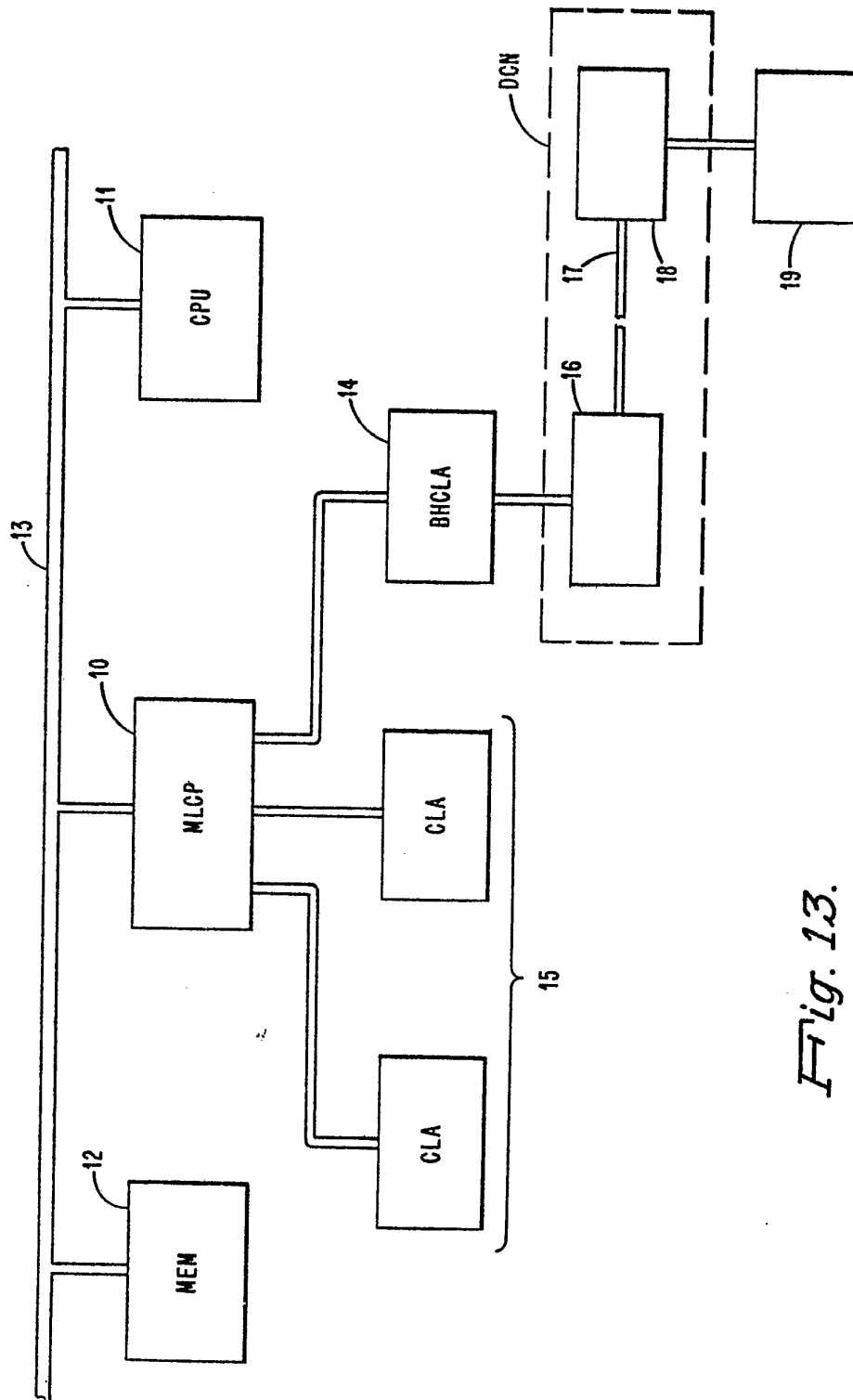


Fig. 12a.



*Fig. 13.*

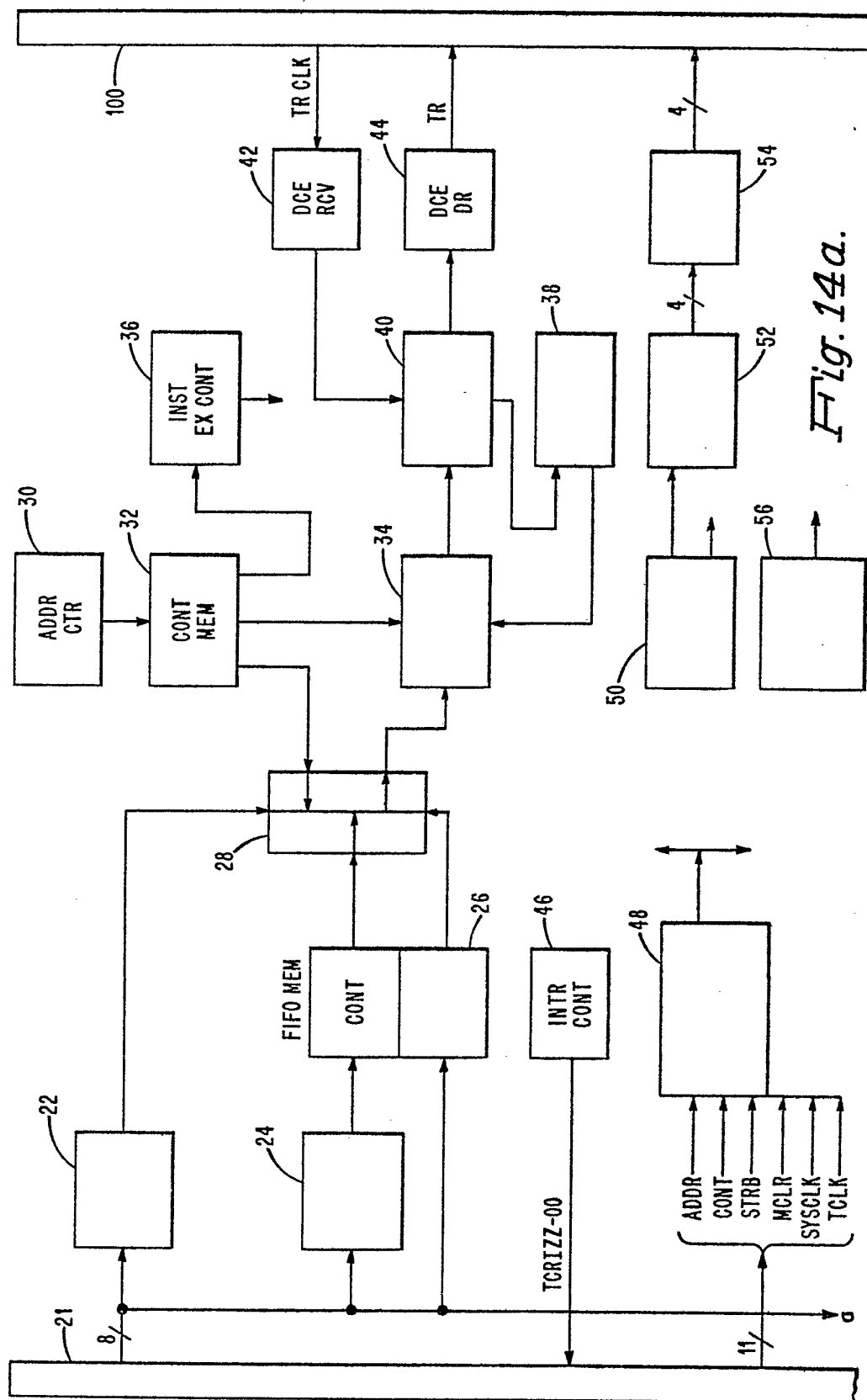
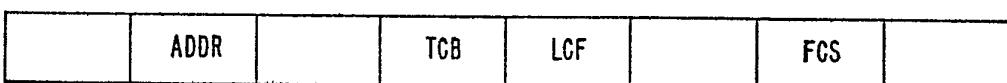
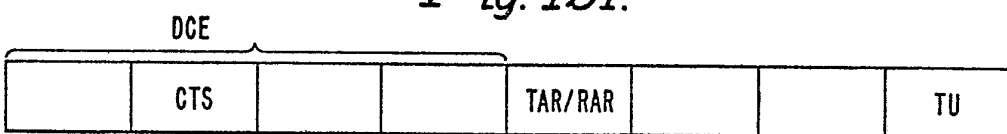
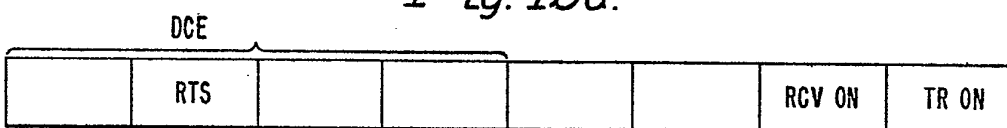
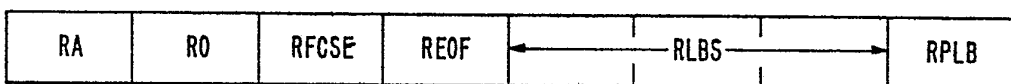
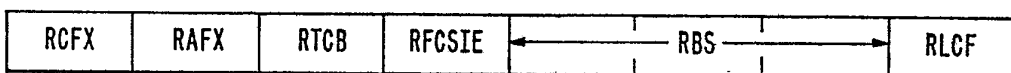
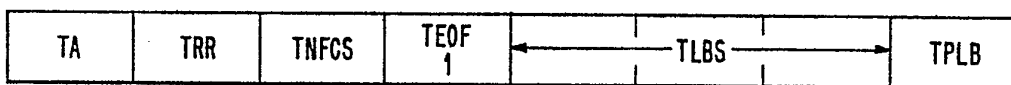
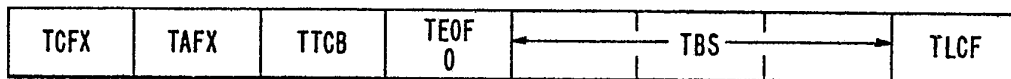
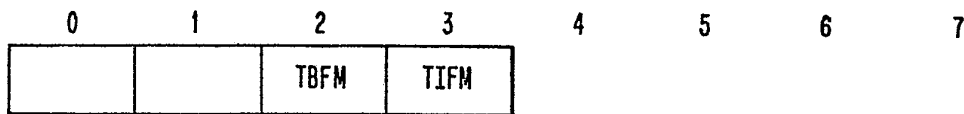
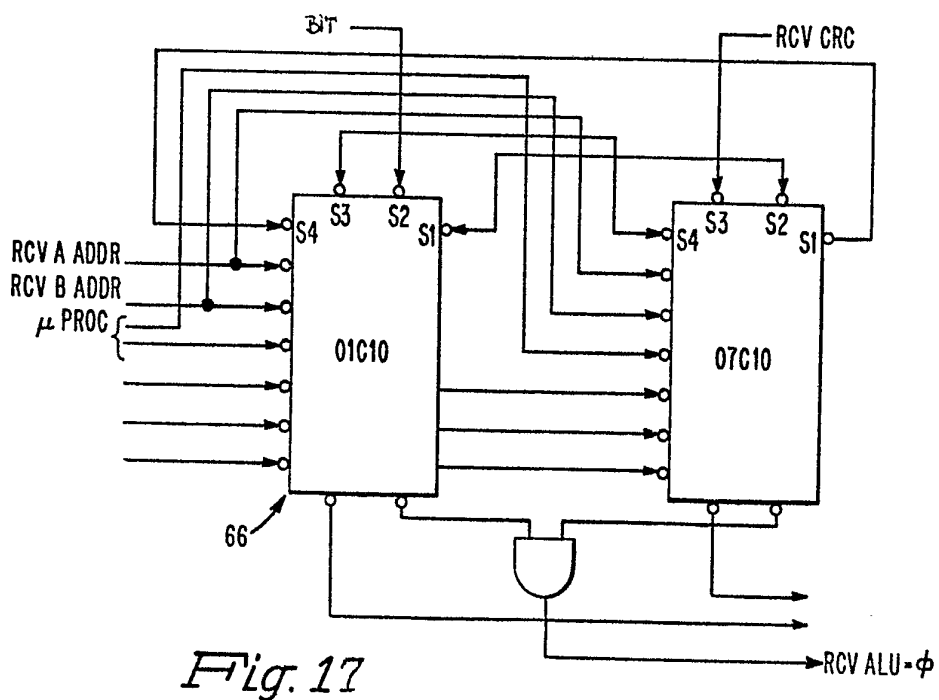
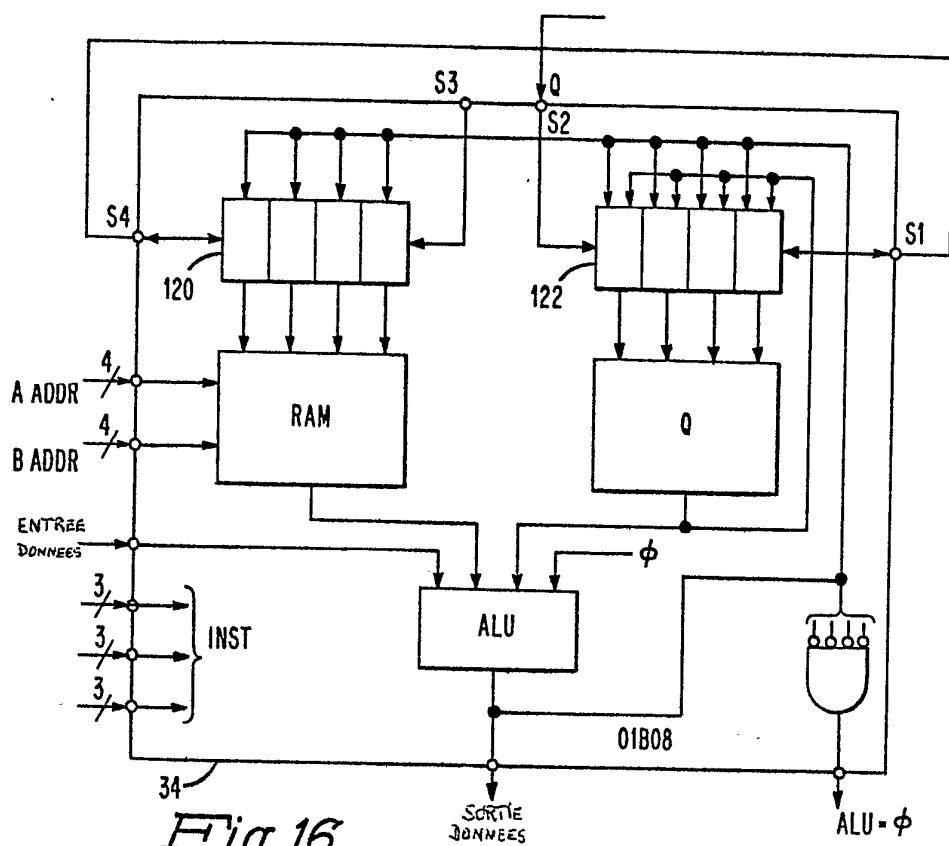


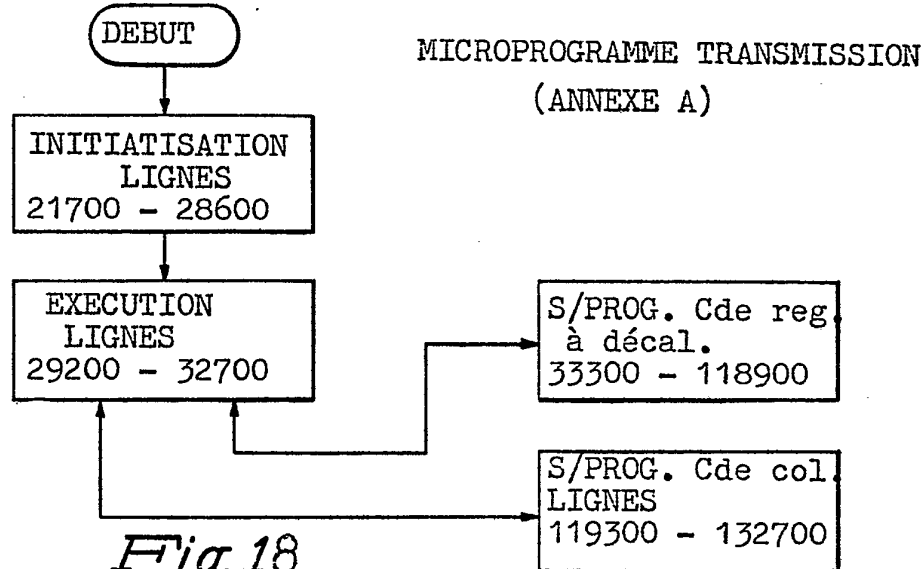
Fig. 14a.



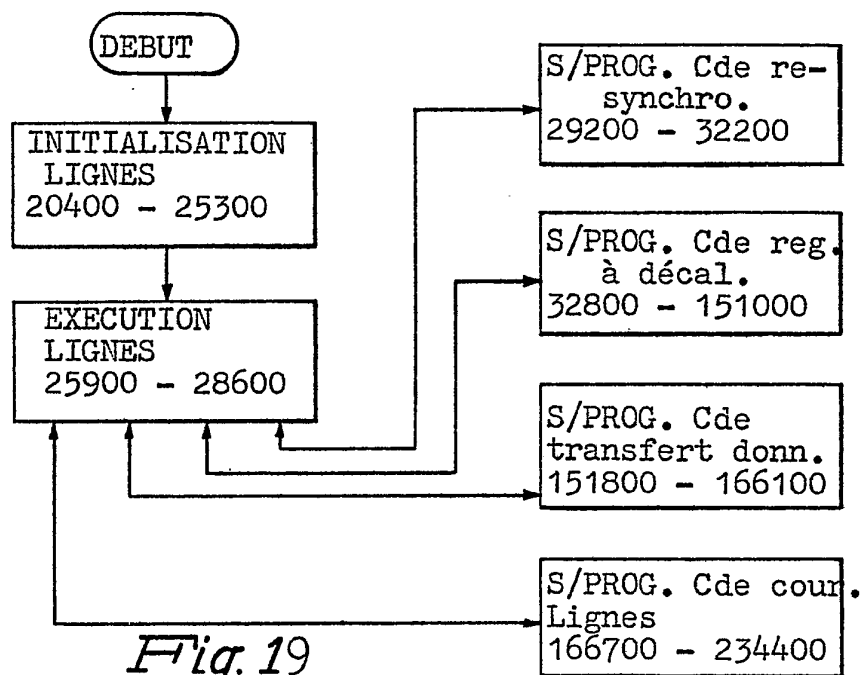


TRANSMISSION
DEFINITIONS
ANNEXE A

LIGNES 1200-20800



RECEPTION
DEFINITIONS
ANNEXE B - 1200
- 19300



ADDRS1-00	01B01	DCEDSR+20	06B05	LADAT4+00	05D04	RALUEZ+01	01C10	RFMDZ3+00	06B12
ADDRS1+10	07A02	DCEDTR-00	08B04	LADAT4+05	04B04	RALUEZ+02	07C10	RFMDZ4+00	08B12
ADDRS2-00	01B01	DCEDTR+00	08B04	LADAT4+17	06C12	PARCZ2+00	11B12	RFMDZ5+00	08B12
ADDRS3-00	01B01	DCEDTR+10	01C05	LADAT5+00	06D04	RCLUKZ-10	12D04	RFMDZ6+00	08B12
ARZZZZ-00	03A04	DCERCK+00	02D01	LADAT5+05	04B04	RCLOKZ+00	11C04	RFMDZ7+00	08B12
ARZZZZ+10	05A04	DCERCK+10	04A05	LADAT5+17	06C12	RCRCDB+00	04C11	RFMIRO+00	01B12
CLKSNR-00	10D01	DCERCK+20	04B05	LADAT6+00	07D04	RCRIXX-00	11C12	RFMIR1+00	03B12
CLKSNT-00	10C01	DCERCK-00	02D01	LADAT6+05	04B04	RCRIXX+00	11C12	RFMIR2+00	06B12
CLKSYN-50	01B01	DCERDZ-10	05A05	LADAT6+17	06C12	RCRIZZ+00	11D12	RFMIR3+00	08B12
CLKSYN+00	01C03	DCERDZ-20	05B05	LADAT7+00	09D04	RCMBZ0+00	11A10	RFMIRZ+00	12B12
CLKSYN+60	01B03	DCERIZ-10	01A05	LADAT7+05	04B04	RCMBZ1+00	01A10	RFMORO+00	01B12
CLKSYN+61	01D03	DCERIZ+00	02D01	LADAT7+17	06C12	RCMBZ2+00	01A10	RFMOR1+00	03B12
CLKSYN+62	02D03	DCERIZ+20	01B05	LR05AD+00	02D02	RCMBZ3+00	01A10	RFMOR2+00	06B12
CNTRL1-00	01B01	DCERTS-00	08B04	LR0ADD-00	01D02	RCMBZ4+00	05A10	RFMOR3+00	08B12
CNTRL2-00	01B01	DCERTS+00	08B04	LR2ADD+00	03B02	RCMBZ5+00	05A10	RFMSZ0+00	01B12
CNTRL3-00	01B01	DCERTS+10	02C05	LR2DTR-00	08A04	RCMBZ6+00	05A10	RFMSZ1+00	01B12
CNTRL4-00	01B01	DCERTS+20	02D05	LR2DTR+00	08A04	RCMBZ7+00	05A10	RFMSZ2+00	01B12
CPDAT0+00	01C01	DCESP2-00	08B04	LR2RTS-00	08A04	RDATAX-00	11C04	RFMSZ3+00	01B12
CPDAT1+00	01C01	DCESP2+00	08B04	LR2RTS+00	08A04	RDATAY-00	02C11	RFMSZ4+00	03B12
CPDAT2+00	01C01	DCESP3-00	08B04	LR2SP2-00	08A04	RDATAY+00	02C11	RFMSZ5+00	03B12
CPDAT3+00	01C01	DCESP3+00	08B04	LR2SP2+00	08A04	RDATAZ+00	02D11	RFMSZ6+00	03B12
CPDAT4+00	01C01	DCETCK+00	02D01	LR2SP3-00	08A04	RDATZE+00	03C11	RFMSZ7+00	03B12
CPDAT5-10	07A04	DCETCK+10	03A05	LR2SP3+00	08A04	RDRDAT+00	02B02	RLR0AD-00	04A02
CPDAT5+00	01C01	DCETCK+20	03B05	LR3ADD+00	07B02	RESYNC-00	07D02	RLR17A-00	04D02
CPDAT6+00	01C01	DCETDZ-10	04C05	LR5ADD-00	03D02	RESYNC+00	07D02	RLR1AD-00	04A02
CPDAT7+00	01C01	DCETDZ-20	04D05	MSTCLR-10	09B02	RFCSCK+00	06C11	RLR2AD-00	04A02
CTSXXX+00	10B06	DIRCON-00	11A04	MSTCLR-11	10B02	RFCSDB-00	12D10	RLR3AD-00	04A02
DECDZ+00	02D01	DIRCON+00	11A04	MSTCLR-30	09D02	RFCSDB-01	04D11	RLR4AD-00	04A02
DECDZ+10	08A05	LADAT0+00	01D04	MSTCLR-50	10A02	RFCSDB-02	07C10	RLR5AD-00	04A02
DECDZ+20	08B05	LADAT0+05	01B04	MSTCLR+00	01B01	RFMBZ0+00	04A12	RLR6AD-00	04A02
DCECTS-30	12C05	LADAT0+17	02C12	MSTCLR+20	09C02	RFMBZ1+00	04A12	RLR7AD-00	04A02
DCECTS+00	02D01	LADAT1+00	02D04	MSTCLR+61	11B02	RFMBZ2+00	04A12	RMSTCL-00	08D02
DCECTS+10	07A05	LADAT1+05	01B04	MSTCLR+62	12B02	RFMBZ3+00	04A12	RONZZZ-00	11A04
DCECTS+20	07B05	LADAT1+17	02C12	PULUPA+00	08B01	RFMBZ4+00	04A12	RONZZZ+00	11A04
DEDECC+11	06C05	LADAT2+00	03D04	PULUPB+00	08B01	RFMBZ5+00	04A12	RQ3R0Z+00	01C10
DEDECC+12	08C05	LADAT2+05	01B04	PULUPX+00	10A05	RFMBZ6+00	04A12	RQ7R0Z+00	07C10
DEDECC+21	06D05	LADAT2+17	02C12	PULUPY+00	11A05	RFMBZ7+00	04A12	RR3R0Z+00	01C10
DEDECC+22	08D05	LADAT3+00	04D04	PULUPZ+00	12A05	RFMDZ0+00	06B12	RR3R0Z+00	01C10
DEDSR+00	02D01	LADAT3+05	01B04	RALUEZ-10	13A11	RFMDZ1+00	06B12	RREADY-00	01C11
DEDSR+10	06A05	LADAT3+17	02C12	RALUEZ+00	06D10	RFMDZ2+00	06B12	RREADY+00	01C11
								RSCMD0-00	08C11

Fig. 20a.

TSTCLZ-00	09A02	TUINS3-00	09B08	WRRFMS+00	08D11
TSTE1R+00	10D01	TUINS4-00	09B08	WRTDAT-10	01C02
TSTE1T+00	10C01	TUINS5-00	09B08	WRTDAT+00	01B02
TSTE2R+00	10D01	TUINS5+10	12C08	WRTFCB-00	06B02
TSTE2T+00	10C01	TUINS6-00	09B08		
TSTEN2-00	02A03	TUINS7-00	09B08		
TSTEN2+10	02B03	TUINS7+10	13C08		
TSTM00-00	11A04	TUINSE-00	09A08		
TSTM00+00	11A04	TUINSE+00	09A08		
TUACC3+00	01A07	TUNDRN-00	06A04		
TUACC7+00	04A07	TUNDRN+00	06A04		
TUCM00+00	10C07	TUPAC0+00	01A07		
TUCM01+00	10C07	TUPAC1+00	01A07		
TUCM02+00	10C07	TUPAC2+00	01A07		
TUCM03+00	10C07	TUPAC3+00	01A07		
TUCM04+00	07C07	TUPAC4+00	04A07		
TUCM05+00	07C07	TUPAC5+00	04A07		
TUCM06+00	07C07	TUPAC6+00	04A07		
TUCM07+00	07C07	TUPAC7+00	04A07		
TUCM08+00	04C07	TUPAC8+00	07A07		
TUCM09+00	04C07	TUPAC9+00	07A07		
TUCM10+00	04C07	TUPI00+00	07A08		
TUCM11+00	04C07	TUPI01+00	05A08		
TUCM12+00	01C07	TUPI02+00	03A08		
TUCM13+00	01C07	TUPI03+00	01A08		
TUCM14+00	01C07	TUPO00+00	01B08		
TUCM15+00	01C07	TUPO01+00	01B08		
TUCM16+00	10D07	TUPO02+00	01B08		
TUCM17+00	10D07	TUPO03+00	01B08		
TUCM18+00	10D07	TXCWB0+00	06A06		
TUCM19+00	10D07	TXCWC1+00	06A06		
TUCM20+00	07D07	TXCWB2+00	06A06		
TUCM21+00	07D07	TXCWB3+00	06A06		
TUCM22+00	07D07	TXCWB4-00	05D08		
TUCM23+00	07D07	WRLCB2-00	07B04		
TUIN23-00	11C08	WRLCBS-00	03C02		
TUIN56+00	07B08	WRRFMB-00	10D11		
TUINS0-00	09B08	WRRFMD-00	09D11		
TUINS1-00	09B08	WRRFMD+00	09D11		
TUINS2-00	09B08	WRRFMS-00	08D11		

Fig. 20 c.