

19 RÉPUBLIQUE FRANÇAISE
 INSTITUT NATIONAL
 DE LA PROPRIÉTÉ INDUSTRIELLE
 PARIS

11 N° de publication :
 (à n'utiliser que pour les
 commandes de reproduction)

2 730 836

21 N° d'enregistrement national :

89 11174

51 Int Cl⁶ : G 06 F 15/82, 13/38

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 23.08.89.

30 Priorité :

43 Date de la mise à disposition du public de la demande : 23.08.96 Bulletin 96/34.

56 Liste des documents cités dans le rapport de recherche préliminaire : *Ce dernier n'a pas été établi à la date de publication de la demande.*

60 Références à d'autres documents nationaux apparentés :

71 Demandeur(s) : IBM CORP INTERNATIONAL
 BUSINESS MACHINES CORPORATION SOCIETE
 DE DROIT DE L ETAT DE NEW YORK — US.

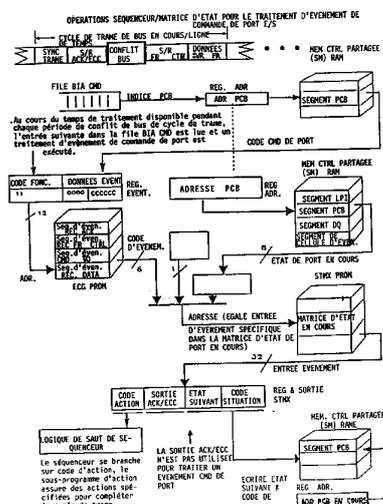
72 Inventeur(s) : BONAFFINI ANDREW MARTIN,
 BONAFFINI KATHLEEN FRANCES, BUEHLER
 MICHAEL JOHN, MILLER HUBERT ALAN,
 PLUNKETT GALEN, RUDOLPH SIDNEY FENTON,
 SWEENEY MICHAEL ARTHUR et WALLIS DONALD
 EARL.

73 Titulaire(s) :

74 Mandataire : CABINET LAVOIX.

54 SEQUENCEUR DE COMMANDE EN TEMPS REEL AVEC LOGIQUE DE MATRICE D'ETAT.

57 Séquenceur de commande en temps réel à hautes performances comprenant une logique de matrice d'état. Ce séquenceur assure une résolution rapide de transitions d'état traitées et les actions de commande requises en fonction d'événements extérieurs détectés et de l'état de processus de commande en cours. Les micros-instructions du séquenceur de commande présentent des données d'événement et d'état en cours comme des entrées à une logique de matrice d'état et déclenche les opérations de commande suivant, les actions de commande de processus requises à réaliser par le micro-code du séquenceur de commande, un état de processus et des données de sortie de commande ou de réponse à un événement. Ce processeur assure une plus grande souplesse pour reconfigurer des combinaisons d'événement à détecter et de réponses désirées. Il assure une résolution plus rapide de transitions d'état de processus de commande et les actions de commande requises et réalise celles-ci avec un répertoire plus important de combinaisons d'événements et de réponses stockées dans une zone de mémoire plus petite que ce qui était nécessaire dans l'état de la technique.



FR 2 730 836 - A1



La présente invention est relative d'une façon générale au traitement de données et concerne plus particulièrement le traitement de données en temps réel, commandé par événement.

5 Le traitement de données commandé par événement trouve des applications dans la commande de processus industriels, la commutation de communications, le traitement de données distribuées, la commande de moteurs à combustion interne, la
10 détection et la prévention de collisions et dans de nombreux autres domaines. Ces applications ont en commun la détection d'événements en temps réel et le calcul de réponses à des combinaisons d'événements au moyen d'un processeur de données à programme
15 enregistré. Un processeur en temps réel commandé par événement porte plusieurs programmes mémorisés, chacun desquels est constitué par une séquence d'instructions qui permet la classification de combinaisons d'événements qui se produisent et qui fournissent des
20 réponses prédéterminées. Ceci nécessite la mise en séquence au moyen d'une série appropriée d'instructions de programme pour assurer la réponse désirée. Un exemple d'un tel processeur de données en temps réel commandé par événement serait un processeur
25 d'appels téléphoniques dans un bureau téléphonique secondaire privé qui détecte les demandes de service et répond par des opérations d'établissement d'appels appropriés.

30 Le problème lié au processeur de données en temps réel commandé par événement typique de l'état antérieur de la technique et le manque de souplesse pour permettre aux responsables du système de modifier des combinaisons d'événements à contrôler et des séquences d'instructions qui doivent être assurées en réponse à ces combinaisons. Dans les applications de

commande de processus, des problèmes ont été rencontrés dans l'exécution d'une résolution rapide de transition d'état de commande de processus. En outre, afin de disposer d'un répertoire adéquat de combinaisons d'événements et de séquences de réponses désirés, de grandes quantités de données doivent être mémorisées en associations avec le processeur de données.

L'invention a donc pour but de fournir un processeur de données en temps réel, commandé par événement perfectionné.

L'invention a également pour but de fournir un processeur de données en temps réel commandé par événement perfectionné qui soit plus souple pour assurer une reconfiguration de combinaisons d'événements et des réponses désirées que les appareils disponibles de l'état de la technique.

L'invention a encore pour but de fournir un processeur de données en temps réel commandé par événement perfectionné qui assure une résolution plus rapide des transitions d'état de commande de processus et les actions de commande requises que les appareils disponibles de l'état de la technique.

L'invention a de plus pour but de fournir un processeur de données en temps réel commandé par événement perfectionné qui soit capable d'emmagasiner un large répertoire de combinaisons d'événements et de réponses désirées avec une allocation de mémoire moins importante que celle qui était nécessaire dans l'état de la technique.

Ces buts ainsi que d'autres, les caractéristiques et les avantages sont réalisés par l'invention décrite ici. On décrit un séquenceur de commande en temps réel à performances élevées qui comprend une

logique à matrice d'état unique. Ce séquenceur de commande en temps réel assure une résolution rapide de transition d'état de commandes de processus et les actions de commandes requises en fonction d'événements extérieurs détectés et de l'état du processus de commande en cours. Les micro-instructions du séquenceur de commande présentent des données d'événement et d'état en cours en tant qu'entrée d'une logique à matrice d'état et déclenche des opérations de matrice d'état. La matrice d'état, à son tour délivre à sa sortie des données définissant et déclenchant l'état suivant de processus, des actions de commandes de processus nécessaires qui doivent être assurées par le micro-code de commande de séquenceur, l'état du processus, et des données de sortie de réponse ou de commande d'événement. Le processeur de données en temps réel commandé par événement suivant l'invention assure une plus grande souplesse pour réaliser une reconfiguration de combinaisons d'événement à détecter et des réponses désirées. Il assure une résolution plus rapide des transitions d'état de commande de processus et des actions de commande requises et il accomplit ceci avec un plus grand répertoire de combinaisons d'événement et de réponses mémorisées dans une zone de mémoire plus petite que ce qui a été nécessaire dans l'état de la technique.

Un aspect important de l'invention est la forte réduction de la complexité de la mise en oeuvre et de la conception qu'elle permet. Des problèmes de commandes d'applications peuvent être définis dans un format tabulaire de niveau élevé qui assure essentiellement sa propre documentation et qui est aisément révisé et entretenu. Etant donné qu'il n'est jamais nécessaire de transférer manuellement la fonction de

commande dans un logiciel, un micro-code ou une circuiterie spécifique, l'invention permet des changements, met à jour et fixe la fonction d'application à mettre en oeuvre aisément. Cette invention facilite
5 une séparation naturelle du mécanisme de mise en oeuvre de la fonction de commande. Des fonctions telles que la transmission de trame de commande, la réception de données, le stockage de données, la mise à jour de tables d'état, etc sont mises en oeuvre en
10 tant qu'opérations séparées autonomes, indépendantes de l'utilisation ultérieure dans la séquence de commande globale. Cette invention permet à un ensemble de mécanismes de commande généralisé d'être spécifié et mis en oeuvre indépendamment de la conception de
15 l'application du système de commande dans son ensemble. La conception des systèmes de commande en temps réel a traditionnellement été l'un des problèmes les plus difficiles et les plus longs dans la science des ordinateurs. Cette invention a montré des perfectionnement
20 radicaux dans le temps global de développement et une réduction des taux d'erreurs dans ces types de problèmes.

Ces objets ainsi que d'autres, les caractéristiques et les avantages de l'invention seront
25 mieux appréciés en se référant aux dessins annexés.

La Fig. 1 illustre quatre différentes applications d'un adaptateur d'Interface de Bus (BIA), qui met en oeuvre l'invention.

La Fig. 2 est un schéma bloc fonctionnel du module adaptateur de Bus d'Interface suivant l'invention.
30

La Fig. 3 est un schéma bloc fonctionnel du module de mémoire partagée (SM).

La Fig. 4 montre le format de commande et de

Contrôle de Bloc de commande d'accès (PCB).

La Fig. 5 illustre des types de trames de bus du réseau de communication en temps réel (RTCN).

La Fig. 6 illustre les codes d'accusés de trames.

La Fig. 7 illustre les champs de commandes de trames des bus RTCN.

La Fig. 8 est une comparaison du séquenceur de commande d'adaptateur d'Interface du Bus RTCN incorporant une logique de matrice d'état.

La Fig. 9 illustre les fonctionnements du séquenceur/matrice d'état pour un traitement d'événement de commandes d'accès E/S.

La Fig. 10 illustre la séparation et le codage général du générateur de code d'événement ECG, PROM pour le traitement d'événement de commande d'accès E/S.

La Fig. 11 est divisée en Fig. 11a, 11b et 11c et illustre le modèle de codage de matrice d'état.

La Fig. 12 illustre les fonctionnements séquenceurs/matrice d'état pour recevoir les événements de code d'accusés de trames.

La Fig. 13 illustre la séparation générale et le codage du générateur de code d'événement (ECG) PROM pour recevoir le traitement d'événement de code d'accusés de trames.

La Fig. 14 illustre les fonctionnements séquenceur/matrice d'état pour recevoir le traitement d'événement de champ de commande trame.

La Fig. 15 illustre la séparation générale et le codage du générateur de code d'événement (ECG) PROM pour recevoir le traitement d'événement de champ de commande de trame.

La Fig. 16 illustre les fonctionnements

séquenceur/matrice d'état pour recevoir le traitement d'événement de données de trame.

La Fig. 17 illustre la séparation générale et le codage du générateur de code d'événement (ECG) PROM pour recevoir le traitement d'événement de données de trame.

La Fig. 18 est la clé pour les diagrammes de transition d'état de niveau haut.

La Fig. 19 est un diagramme de transition d'état pour un état désactivé.

La Fig. 20 est un diagramme de transition d'état pour les opérations de commande de signaux d'échange.

La Fig. 21 est un diagramme de transition d'état pour sous-programme de commandes de signaux de réception.

La Fig. 22 est formée de trois pages des Figs 22a, 22b et 22c qui montrent le code de source PROM de la matrice d'état pour l'état de désactivation.

La Fig. 23 qui est formée des trois Figs 23a, 23b et 23c illustre le code de source PROM de la matrice d'état pour l'état EXSIG-S-WT-REACK-1.

La Fig. 24 qui est formée des trois Figs 24a, 24b et 24c illustre le code de source PROM de matrice d'état pour l'état RSIG-WAIT-SIG-1.

La Fig. 25 qui est formée des Figs 25a, 25b et 25c montre le code de source PROM de la matrice d'état pour l'état RSIG-WT-SIGDATA-1.

La Fig. 26 représente un sommaire de l'utilisation des ressources de la logique de matrice d'état et la mise en oeuvre de l'application RTCN NIU BIA.

Description du mode de réalisation préféré1.1. But

On décrit un séquenceur de commande en temps réel à hautes performances qui comporte une logique de matrice à état unique. Ce séquenceur de commande en temps réel réalise la résolution rapide de transitions d'état de processus de commande et des actions de commande requises en fonction d'événements extérieurs détectés et d'état de processus de commande en cours.

Les micro-instructions du séquenceur de commande présentent des événements et des données d'état en cours en tant qu'entrées à la logique de matrice d'état et initialisent les opérations de matrice d'état. La matrice d'état fournit tour à tour des données définissant et initialisant l'état du processus de commande suivant, des actions de commande du processus requises, destinées à être réalisées par micro-code de séquenceur de commande, des états de processus et des données de sortie de réponse d'événements ou de commande.

Il est commun de fournir une spécification et une définition rigoureuses des processus de commande sous la forme de diagrammes de transmission d'état et/ou de tables/matrices d'état définissant, pour chaque entrée ou événement détecté possible, l'état de transition suivant et les sorties et actions de commande nécessaires. Lorsqu'on met en oeuvre cette spécification de processus de commande dans des séquenceurs/micro-contrôleurs classiques, la spécification de matrice d'état direct et compact doit être transformée en code séquentiel sous la forme d'arbres à branches longues correspondant à toutes les combinaisons d'état en cours et d'événement possible. Dans le séquenceur de commande comportant la logique de

matrice d'état, qui est décrite dans ce document, le processus de commande est codé et chargé directement dans le format de matrice d'état intrinsèque et le séquenceur de commande fonctionne directement sur
5 celui-ci.

Une inspection du code de matrice d'état est une inspection du processus de commande mis en oeuvre en cours.

L'objet de ce document est de décrire les
10 fonctionnements de la logique de matrice d'état pour l'application dans le module d'Adaptateur d'Interface de Bus (BIA) associé avec la mise en oeuvre d'un réseau de communication en temps réel à fibres optiques de haute performance (RTCN). L'approche des
15 concepts et la configuration générale, décrites et réalisées ici, peuvent être appliquées à une large gamme d'applications de séquenceurs de commande avec les mêmes avantages.

1.2. Portée

20 L'organisation, le contenu, et la portée de ce document sont comme suit.

La suite de cette section définit l'exemple d'application RTCN. La configuration de modules fonctionnels au niveau machine/circuiterie de l'unité
25 d'interface du réseau RTCN est définie. Deux de ces modules qui sont associés avec toute description de la logique de matrice d'état sont définis. Ces modules sont les BIA, qui contiennent à la fois le séquenceur de commande et la logique de matrice d'état et le
30 module de mémoire partagée (SM) qui fournit quelques ressources de files et de mémoires de bloc de commande auxiliaire. La mise en oeuvre de la logique de matrice d'état RTCN du processus de commande d'application est comparée avec une mise en oeuvre de l'art antérieur

utilisant trois séquenceurs de commande classiques. L'exemple de mise en oeuvre de l'art antérieur est tiré du Bus de Données de Système réparti (DSDB) décrit dans le brevet U.S. 4,373,183 de Means et al.
5 intitulé "Bus Interface Units Sharing a Common Bus Using Distributed Control for Allocation of the Bus", délivré à IM Corporation. Le RTCN est une évolution de l'architecture et une amélioration au point de vue prix/performance du DSDB.

10 La deuxième section définit les fonctionnements logiques de la matrice d'état général pour l'exemple d'application de processus de commande. Elle couvre les fonctionnements associés aux entrées du type des quatre événements généraux vers la logique de
15 matrice d'état.

Dans la troisième section, une partie du codage de matrice d'état en cours pour l'exemple d'application et ses fonctionnements sont décrits. Cet exemple couvre l'initialisation et l'exécution de
20 l'une des commandes de ports d'entrée/sortie E/S du BIA.

La quatrième section résume les résultats de l'exemple d'application.

1.3 Brevet voisin

25 Brevet U.S. 4,373,183 de Mean et al. intitulé "Bus Interface Units Sharing a Common Bus Using Distributed Control for Allocation of the Bus", au nom d'IBM Corporation.

30 1.4. Application du module d'adaptateur d'interface de bus BI1 du réseau de communication en temps réel (RTCN)

Une configuration d'un séquenceur de commande en temps réel à haute performance comportant une logique de matrice d'état a été utilisée dans la con-

ception et le développement du réseau de communication en temps réel RTCN. Le RTCN est un réseau de communication de zone locale à fibres optiques de haute performance capable de répondre à des besoins d'application en temps réel complexes.

Quatre modules fonctionnels de matériel communs ont été développés à partir desquels une variété d'unités d'interface de réseau, une unité de pont, des unités de passerelle et une unité de contrôle de bus peuvent être définies. Le jeu de modules fonctionnels de matériel commun comprend un module d'adaptateur d'interface de bus (BIA), un module de mémoire partagée (SM), un module de processeur de service E/S (IOSP) et un module d'adaptateur d'interface du dispositif (DIA). Quelques unes des configurations d'unité (RTCN) dérivées de ce jeu de module fonctionnel de matériel commun sont représentées sur la figure 1. Une configuration de séquenceur de commande comportant une logique de matrice d'état a été utilisée dans le module BIA clé. Un ou plusieurs BIA sont utilisés dans toutes les configurations RTCN. Les prototypes des MOD1-NIU, MOD2-NIU et BMU ont été conçus dans une variété de configurations de réseau RTCN. Une description des fonctionnements des matrices d'état/séquenceur de commande incorpore également quelques ressources de file et de mémoire cloisonnées sur le module SM. Une vue générale des modules SM et BIA suit.

1.4.1. Module d'adaptateur d'interface de bus (BIA).

1.4.1.1. Rôle et fonction du BIA.

Le BIA fournit un interface au milieu de communications/bus de fibres optiques. Il fournit la commande de niveau de liens associée à la transmission et à la réception de trames de données et de commandes

de bus. Il fournit un interface de réponses et de commandes aux fonctions de commandes et de réseaux, au-dessus des milieux et du niveau de liens qui réside dans le module IOSP.

5 1.4.1.2. Schéma synoptique et vue générale du niveau machine.

La Fig. 2 est un schéma synoptique fonctionnel du module BIA du RTCN. Une brève description des ressources réparties dans ce module suit.

10 1.4.1.2.1. Interface de fibres optiques.

La logique d'interface de fibres optiques fournit la logique d'horloge et de codage et décodage et d'émission et de réception pour l'interface de milieu de transmission optique. La logique de réception et d'émission double est prévue pour des questions de disponibilité/accessibilité. Le BIA peut commuter sur l'un ou l'autre de jeux d'émetteur/récepteur. La logique d'interface de fibres optiques est commandée par des signaux engendrés par le séquenceur de commande du BIA.

15 20 1.4.1.2.2. Données FIFO reçues du bus et transmises au bus.

Ces tampons de données premier entré, premier sorti (FIFO) mettent en tampons les données en cours reçues du bus et transmises au bus, respectivement. Le code de matrice état/séquenceur de commande BIA commande les transferts de données entre ces FIFO de données et des tampons de fibres de données (DQ) des blocs de commandes (PCB) ou des cellules d'événement dans le module SM en fonction du BIA commandé, du champ de trame, du type de trame et de l'opération BIA commandée.

30 1.4.1.2.3. Logique de générateur de signature.

Cette logique réalise "un algorithme de

division polynomial 16- CRC" sur lequel toute
détection d'erreurs de données de bus de RTCN est
basée. Lorsque des données sont reçues elles sont
dirigées sur cette logique et une signature est cal-
culée, et comparée à une signature reçue attachée aux
5 données. Sur des opérations d'émission les données
sont dirigées vers cette logique et une signature est
calculée qui est jointe à la donnée transmise.

1.4.1.2.4. ALU et séquenceur de comande.

10 Le micro-code s'exécutant dans le séquenceur
de commande en coopération avec la logique de commande
de matrice d'état unique, commande toutes les opéra-
tions du BIA. Le micro-code de séquenceur réalise
toutes les opérations d'émission de réception et de
15 synchronisation de cycle de trame en temps réel. En
différents points de chaque cycle de trame il présente
des données d'événements internes ou de cycle de trame
à la logique de commande de matrice d'état qui résoud
l'événement en fonction de l'état du port en cours et
20 produit des sorties qui dirigent les opérations ulté-
rieures de commande de cycle de trame de séquenceur.
Ces fonctionnements de matrice d'état généraux sont
définis dans la section 2. L'architecture du séquen-
ceur de commande et le jeu de micro-instructions
25 aident uniquement la capacité des séquenceurs à entrer
des données d'état en cours et d'événements dans la
logique de matrice d'entrée, initialisent les fonc-
tionnements de la matrice d'état et à être dirigés par
les sorties de matrice d'état. La sortie de code
30 d'action de logique de matrice d'état est dirigée vers
la logique de sauts des séquenceurs.

1.4.1.2.5. Mémoire de commande.

Une PROM de 8K par 16 bits contient le mi-
cro-code du séquenceur de commande.

1.4.1.2.6. Logique de commande de matrice d'état.

A quatre points spécifiques dans chaque cycle de commande de trame (voir 1.4.3) le séquenceur de commande peut présenter des données d'événement à la logique de commande de matrice d'état. Les quatre types d'événements sont l'événement ECC/ACK de trame reçue, un événement de décalage ou de commande du port DIA, un événement de champ de commande de trame reçue et un événement de données de trame reçue. La donnée d'événement de rangées est compressée en un code d'événement à 6 bits par la PROM de générateur de code d'événement de matrice d'état (EGC). L'état de port en cours (8 bits) est lu du bloc de commande de port (PCB) (du port associé) de la mémoire partagée dans le module SM/DIA. L'état du port en cours et le code d'événement sont combinés pour former une adresse de PROM de matrice d'état pour accéder à l'entrée d'événement spécifique dans la matrice d'état spécifique. La sortie de la PROM de matrice d'état dirige les fonctionnements ultérieurs de commande de cycle de trame de micro-code du séquenceur de commande pour le cycle de trame en cours. Les sorties de la PROM de matrice d'état consistent en un état de port suivant, un code d'action, un code d'état de port et un code ACK/ECC de trame. La sortie d'état suivante remplace l'état en cours lu précédemment dans le PCB du port. La sortie du code d'action indique au micro-code de séquenceur comment procéder dans le cycle de trame en cours. La sortie d'état de port peut être stockée dans le PCB et un signal de réponse de commande peut être envoyé au système de fonctionnement du réseau (NOS) qui réside dans le module IOSF en fonction du code d'action spécifique. La sortie de code ACK/ECC de trame peut-être un cycle de trame transmis suivant,

dépendant une fois de plus de la sortie de code d'action spécifique.

1.4.1.2.7. Fichier de registre.

Huit registres polyvalents sont prévus pour
5 une utilisation par le séquenceur de commande.

1.4.1.2.8. Registres de données et d'adresses de mémoire.

Le registre d'adresses de mémoire (MAR) et le registre de données de mémoire (MDR) donnent accès
10 à une mémoire partagée et à des registres et des files configurées de mémoire sur le module SM.

1.4.1.2.9. Registre ID de noeuds et de domaine de bus.

Ce registre autorise le micro-code du séquenceur à accéder à son propre domaine de bus et aux
15 informations ID de noeuds qui sont câblés par le panneau arrière du NIU.

1.4.2. Module de mémoire partagé (SM)

1.4.2.1. Rôle et fonctions

Le module SM fournit une mémoire de tampons
20 de données, une file FIFO et de priorité et des ressources d'horloge en temps réel qui sont partagées avec le module IOSP et le module DIA. Tous ces éléments sauf l'horloge en temps réel sont auxiliaires aux fonctionnements de matrices d'état généraux
25 réalisés par le module BIA.

1.4.2.2. Schéma synoptique et vue générale au niveau machine.

La Fig. 3 est un schéma synoptique fonctionnel du module SM. Une brève description uniquement
30 des ressources SM qui sont auxilliaires aux fonctionnements de matrices d'états généraux suit.

1.4.2.2.1. File de contenu de bus.

Cette file de priorité de matériel contient le port ID (Index de Bloc de Commande de Port PCB) et

la priorité des ports dont les fonctionnements sont en attente sur la transmission d'une trame. Seul le mi-cro-code de séquenceur de BIA peut lire ou écrire cette file.

5 1.4.2.2. File de commande de BIA.

Cette file de priorité de matériel contient le port ID (Index PCB) et la priorité de port avec une commande de port en attente du NOS. Le micro-code de séquenceur de BIA peut uniquement lire cette file, ce
10 qu'il fait durant les fonctionnements de matrices d'état de traitement de commande de BIA. Le NOS peut uniquement écrire cette file.

1.4.2.2.3. File de répartition d'étape de travail.

Cette file de priorité de matériel contient
15 le port ID (Index PCB) la priorité et l'ID étape de travail du NOS des fonctionnements de commande de port qui ont été terminés par le BIA. Cette file peut uniquement être écrite par le BIA, comme dirigé par le code d'action de sortie de la logique de matrice
20 d'état et peut uniquement être lue par le NOS.

1.4.2.2.4. File d'allocation de cellule d'événement.

Cette file FIFO de matériel contient l'ID des cellules d'événement disponibles (tampon de donnée de trame de signal dans le SM) utilisable par le mi-
25 cro-code du BIA. Le micro-code de séquenceur du BIA peut soit lire ou écrire cette file sous la direction du code d'action de sortie de la logique de matrice d'état. Le NOS peut uniquement écrire cette file.

1.4.2.2.5 File d'allocation DQ.

30 Cette file FIFO de matériel contient l'ID des tampons (DQ) de files de données disponibles utilisables par le micro-code du BIA pour recevoir des paquets de données reçues. Le micro-code du séquenceur du BIA peut soit lire, soit écrire cette file sous la

commande du code d'action de sortie de la logique de matrice d'état. Le NOS peut uniquement écrire cette file.

1.4.2.2.6. Registre de commande.

5 Ce registre contient les différents bits de commande NIU. Le micro-code BIA et le NOS peuvent soit lire soit écrire ce registre.

1.4.2.2.7. Registre d'état.

10 Ce registre contient plusieurs bits/conditions d'état NIU. Le micro-code BIA et le NOS peuvent uniquement lire ce registre.

1.4.2.2.8. Mémoire de commande partagée (RAM).

15 La mémoire de commande partagée contient 128K mots de 16 bits. Elle peut être lue ou écrite soit par le micro-code BIA ou le module IOSP ou le module DIA. Elle est divisée en segments de blocs de commandes et de tampons de données suivants :

20 1. Segment de mémoire LPI - Ce segment contient 64 K mots. Chaque mot est associé avec les commandes l'activation et la désactivation d'une adresse de réseau virtuel (Identificateur de trajet logique (LPI)).

25 Chaque mot contient un état d'activation et de désactivation et l'index du PCB associé. Le séquenceur BIA lit ce segment durant le traitement de réception d'un événement de commande de trame.

2. Segment de contrôle de port PCB - Ce segment contient 8K mots divisés en 128 blocs de commandes de Port E/S avec 64 mots chacun.

30 Chaque PCB est associé à la commande de l'un des 128 fonctionnements de Port E/S concurrents initialisés par le NOS dans le module IOSP.

Chaque PCB est divisé en 4 segments de 16 mots. Ces segments sont le segment de commande, le

segment de tampon de trame reçue, le premier segment de tampon de transmission de trame et le second segment de tampon de transmission de trame. L'état du port en cours et le code de commande du port sont
5 maintenus dans les segments de commande du PCB et sont accédés durant des fonctionnements de la logique de matrice d'état. Les sorties de code d'état et d'état suivants, provenant du fonctionnement de la matrice d'état sont écrites dans les segments de commande et
10 de contrôle par le séquenceur de commande sous la commande du code d'action de sortie de logique de matrice d'état. Le format des segments de commande PCB est représenté sur la Fig. 4.

3. Segment de tampon de file de données DQ
15 (ce segment contient 47 K mots divisés en 47 tampons DQ avec 1K mot chacun (2K multiplats)).

Chaque tampon PQ maintient un paquet de données reçues du bus RTCN ou en attente d'être transmis sur le bus.

20 4. Segment de cellules d'événement -ce segment contient 2K mots divisés en 128 cellules d'événement avec 16 mots chacune. Les cellules d'événement sont utilisées pour mettre en tampon des trames reçues sur des Ports qui sont dans un état de
25 commande "Contrôle".

5. Segment de commande fixe -ce segment contient 1008 mots de paramètres de commandes fixes.

30 6. Segment de bloc de commande DIA et de bloc de commande d'utilisateur (UCB) -ces segments de bloc de commande contiennent 2 K et 4 K mots respectivement et utilisés uniquement par le DIA et le IOSP.

1.4.3. Types de trame de bus RTCN et traitement général de ligne de temps de cycle de trame.

1.4.3.1. Types de trame comme représenté sur

la Fig. 5, quatre types de trames sont transmis sur le bus RTCN. Ceux-ci sont les trames nulles, les trames de jetons, les trames de signal et les trames de BAL et les paquets de données associées.

5 Les trames nulles.

Les noeuds RTCN sont en conflit pour les fentes de trames sur le bus lorsqu'ils ont des informations ou des données de commande à transmettre. Une trame nulle apparaît lorsqu'aucun noeud n'est en
10 conflit pour une fente de trame donnée.

Trame de jeton.

Un noeud qui reçoit des données sous un protocole de transaction de message périodique envoie une trame de jeton lorsqu'il est prêt pour recevoir le
15 prochain paquet de données/trame de BAL.

Trame de signal.

Les trames de signal sont transmises dans différents protocoles de transmission de message pour présenter des informations de réponse, de commande, de
20 niveau de transaction et de message. Ils sont également transmis pour réaliser plusieurs fonctions de commandes de réseaux.

Trame de BAL/(paquet de données).

Les trames de BAL sont utilisées dans plu-
25 sieurs protocoles de niveaux de transaction/message pour envoyer des données d'entrée/sortie d'utilisateur. Chaque trame de BAL et un paquet de données associé peut transmettre de 128 à 2K multiplats de données et une transaction/message unique peut con-
30 tenir un ou plusieurs paquets de données/trames de BAL.

1.4.3.2. Champs/segment de cycle de trame commun.

Comme représenté sur la Fig. 5 les trois

premiers champs/segments de cycle de trames sont communs à et associés avec tous les quatre types de trames.

Synchronisation de trames.

5 Pour le début de chaque cycle de trame, le noeud de bus qui est le générateur de synchronisation de bus en cours (noeud BSG) transmet une configuration de synchronisation et tous les autres noeuds (à l'écoute) tentent d'établir cette synchronisation.

10 Accusé de réception de trame.

A la suite de l'établissement de la synchronisation de trame le noeud qui reçoit la trame précédente (à condition qu'elle n'ait pas été envoyée en un mode de diffusion) transmet un code d'accusé de réception de trame (ACK/ECC). Celui-ci est un code de correction d'erreurs à sept bits (un code ACK à trois bits plus quatre bits ECC). Celui-ci fournit une correction d'erreurs de bits unique et une détection d'erreurs de bits doubles ou triples. Les codes d'accusés de réception de trames sont représentés sur la Fig. 6.

15

20

Conflit de bus.

Après la transmission de l'accusé de réception de trame antérieur, tous les noeuds de bus, avec information à transmettre, seront en conflit pour l'intervalle de trame en cours. Chaque noeud en conflit transmet simultanément un code de priorité à huit bits suivi de son noeud ID physique à six bits ("Adresse de conversation TA"). La transmission commence avec le bit de priorité le plus significatif. Après chaque transmission de bit, chaque noeud reçoit le bit résultant "Dot ORed". Si le bit reçu ne correspond pas au bit transmis par un noeud, il sort du processus de conflit. Le noeud qui reçoit une correspon-

25

30

dance sur les quatorze bits est le gagnant du conflit. Il est à noter que l'adresse de conversation agit comme un sélecteur de décision si plus d'un noeud a la même plus haute priorité. Si aucun noeud n'est en
 5 conflit pour l'intervalle de trame en cours, une trame nulle est déclarée et l'intervalle de trame suivant est commencé avec la même synchronisation de trame. Champ de commande de trame.

Après le conflit de bus, le noeud vainqueur
 10 du conflit transmet des champs de commande de trame définissant la trame et son adressage d'un noeud cible et de trajet. Les champs de commande des trames, comprenant cinq mots de seize bits, sont représentés sur la Fig.7.

15 Données de trame.

Après la transmission des champs de commande de trame, des données de trame sont transmises à la fois dans des cycles de trame BAL et de signaux, comme décrits par la suite.

20 Données de signaux.

Seize multiplats (huit mots) de données de signaux sont transmis dans chaque trame de signal. Ces données sont de manière générale des paramètres associés avec des réponses de commande/commande de réseau,
 25 immédiat ou des réponses de commande/commande de niveau de protocole de message/transaction NOS. Paquets de données de trame BAL.

Le cycle de paquets de données/trame de BAL est de longueur variable. Des données d'entrée/sortie
 30 E/S d'utilisateur, pouvant aller de 128 à 2 K multiplats, par incréments de 128 multiplats, peuvent être transmises dans chaque cycle de paquet de données/trame BAL donné.

Le compte de sous-paquet dans les champs de

commande de trame spécifie la quantité de données transmises dans le cycle de trame associée.

Signature de données de trame.

Après la transmission de données de trame, un mot de signature de données de trame à seize bits est transmis. Ce mot est engendré par le BIA transmettant en réalisant un "algorithme de division polynomiale par 16 - CRC " sur le paquet précédent de données/trame de BAL ou de données de signal.

Le BIA récepteur engendre également une signature de données de trame sur des mots de données de trame reçus et la compare avec la signature de données de trame qui a été transmise.

1.4.3.3. Points dans les cycles de trame de bus RTCN auxquels des fonctionnements de logique de matrice d'état sont réalisés.

Comme cela est représenté sur la Fig.5, il existe quatre points dans un cycle de trame donné où le séquenceur de commande peut initialiser un fonctionnement de logique de matrice d'état. Chaque point constitue un type d'événement donné. Durant la transmission de l'accusé de réception de trame, si le BIA a transmis une trame pendant le cycle de trames antérieur, le séquenceur de commande reçoit le code d'accusé de réception plus le ECC et les présente à la logique de matrice d'état pour une résolution d'un événement de code d'accusé de trame reçue. Pendant le temps disponible, lors de la période de conflit de cycles de trame de conflit de bus, le séquenceur de commande teste la file de priorité de commande du BIA, sur le module SM, pour voir si elle a une ou plusieurs entrées.

Si la file maintient une entrée, le séquenceur lit une entrée (de plus haute priorité) de la

file. Il lit le code de commande du BIA dans le PCB (dans le module SM) indiqué par l'index fourni dans l'entrée de file et le présente à la logique de matrice d'état pour une résolution d'un événement de commande de port entrée/sortie. Pendant la période de champ de commande de trame, si le BIA n'est pas en conflit ou perd un conflit de trame de bus, pendant le segment de conflit de trame de bus précédent, le séquenceur de commande reçoit les données de champ de commande de trame. Il utilise le paramètre LPI de cible reçue pour lire la mémoire LPI (dans le module SM). Si la cible LPI est validée sur le noeud, il présente les données de champ de commande de trame reçue à la logique de matrice d'état pour une résolution d'un événement de champ de commande de trame reçue. Si le code d'action de sortie de la logique de matrice d'état, à partir du fonctionnement de l'événement de champ de commande de trame reçue précédent, dirige le séquenceur de commande pour recevoir les données de trame, le séquenceur de commande va pour recevoir les données. Il validera le générateur de signature et assurera le transfert des données au tampon des données spécifiées par le code d'action de sortie de logique de matrice d'état antérieur. Il présente la condition de données résultantes à la logique de matrice d'état pour une résolution d'un événement de données de trame reçue.

1.5 Art antérieur et comparaison avec une mise en oeuvre utilisant un séquenceur de commande incorporant une logique de matrice d'état.

La Fig.8 fournit une comparaison entre la mise en oeuvre de l'exemple du processus de commande milieu/niveau de liaison utilisant le séquenceur de commande comportant une logique de matrice d'état et

une mise en oeuvre de l'art antérieur. La mise en oeuvre de l'art antérieur concerne le bus de données de système réparti développé antérieurement (DSDB). Le RTCN est une évolution de l'architecture et une amélioration au point de vue performance/coût du DSDB.

5 Les architectures des bus de base pour les deux réseaux sont très similaires. Dans la partie supérieure de la Fig.8, les configurations de modules fonctionnels pour les RTCN et DSDB NIU sont représentées et comparées.

10 Dans les deux cas, les modules fonctionnels associés avec les commandes de milieu/niveau de lien sont indiqués. Dans la partie intermédiaire de cette figure, les ressources de stockage et de processeur de commande sur ces modules sont représentées et comparées.

15 Un résumé des ressources nécessaires est fourni dans la partie basse de cette figure. Dans la mise en oeuvre d'un RTCN, le processeur de commande unique comprenant et complété par la logique de matrice d'état réalise la fonction de commande d'application qui nécessite trois microprocesseurs/séquenceurs de commande classiques pour réaliser la mise en oeuvre du DSDB.

20 Un examen attentif du microcode dans les microprocesseurs classiques dans les modules BBS et DSDB CPI montre que les processeurs sont appliqués dans un rapprochement entre les événements détectés et l'état du processus de commande en cours. Les jeux de code d'application dans ces deux microprocesseurs sont pour la plupart des arbres à commande à branches séquentielles importants. Le problème de commande de l'application intrinsèque est la mise en oeuvre d'une

25 matrice d'état qui est mieux réalisée par une logique qui fonctionne ou émule à peu près comme une matrice d'état. C'est ce que fait le séquenceur de commande unique de mise en oeuvre d'un RTCN comportant une

30

logique de matrice d'état.

Deuxième section - Fonctionnement de matrice d'état général.

2.1 - Initialisation du SM et du BIA.

5 Pendant l'initialisation du NIU, le micro-code du séquenceur de commande du BIA réalise les opérations suivantes.

10 Il établit le paramètre d'état en cours dans tous les PCB (sur le module SM) à l'état désactivé. Il rétablit et efface la file de commande du BIA, la file de conflit de bus et la file de répartition d'étape de mots (toutes sur le module SM). Il rétablit et efface toutes les entrées LPI dans la mémoire LPI (également sur le module SM) à l'état désactivé. En résultat, au-

15 cune émission ou réception de trame ni aucun traitement d'événement de matrice d'état ne se produit tant que le NOS dans le module IOSP délivre une commande de port E/S en entrant celle-ci dans un PCB et en écrivant l'index de PCB dans la file de commande de BIA.

20 Le premier événement que le séquenceur de commande détecte et présente à la logique de matrice d'état est l'événement de commande de port E/S.

2.2 - Traitement de l'événement de commande de port E/S.

25 La Fig.9 illustre le traitement d'événement de commande de port E/S qui est décrit ci-dessous.

2.2.1 - Détection d'événement et ensemble de paramètres d'entrée de matrice d'état.

30 Pendant le temps disponible, au cours de la période de conflit de bus dans un cycle de trame, le séquenceur de commande teste la file de commande du BIA (sur le module SM) pour une entrée. Si la file n'est pas vide, il lit une entrée dans la file (la

plus haute priorité). L'entrée contient un index de PCB. Le séquenceur convertit l'index en l'adresse du segment de commande du PCB. Il lit le paramètre d'état de port en cours et le code de commande du BIA à partir du PCB (dans la mémoire partagée sur le module SM).

2.2.2 - Compression de données d'événement, d'entrée et de sortie d'une PROM du générateur de code d'événement de matrice d'état (ECG).

Le séquenceur établit le code de fonction pour l'événement de commande de port E/S dans les registres d'événement de la logique de matrice d'état. Il transfère le code de commande du BIA (lu dans le PCB) aux registres d'événement. Le code de fonction établi précédemment entraîne l'ensemble de données d'événements approprié dans les registres d'événements, hors, du bus de données du BIA interne comme représenté sur la Fig.9. Ceci initialise un accès de lecture de la PROM du générateur de code d'événements (ECG) de la logique de matrice d'état. La sortie de la PROM ECG est un code d'événement à six bits qui est dirigé dans les bits 1 à 6 de la PROM de matrice d'état. La subdivision de la PROM ECG, le codage du segment d'événements de commande de port E/S et la compression de données d'événements de commande de port E/S résultante est représenté sur la Fig.10. Ainsi qu'on peut le voir sur la Fig.10, le code de fonction dans le format d'entrée divise la PROM ECG en segments de 1 K pour chaque type d'événement. La PROM physique utilisée dans le prototype (8K x 8) a deux fois la taille nécessaire (4K x 8). Le fonctionnement de la PROM ECG comprime les 64 combinaisons de données d'entrée d'événement de commande de port E/S, en 19 codes d'événement à six bits. Le fonctionnement de la

PROM ECG fournit une détection de commande invalide et produit un code d'événement de sortie "commande invalide". Les positions de PROM ECG invalides ou inutilisées, dans le segment d'événement de commande, sont codées pour produire un code d'événement de sortie "position ECG invalide". Sans la compression de données fournie par l'étage de la PROM ECG de la logique de matrice d'état, la PROM de matrice d'état aurait seize fois sa taille actuelle (8 mégabits à la place de 512 K bits).

2.2.3 - Entrée et sortie de la PROM de matrice d'état.

Le code d'état en cours à huit bits, lu dans le PCB (2.2.1 plus haut) est transféré par le séquenceur aux huit bits d'ordre élevé du registre d'état en cours, de la logique de matrice d'état pour initialiser le fonctionnement de la PROM ECG. La sortie de code d'événement à six bits de la PROM ECG est concaténée avec les six bits d'ordre faible du registre d'état en cours initialisant un accès de lecture de la PROM de matrice d'état. L'entrée de code d'événement à 32 bits spécifique dans la matrice d'état associée à l'état en cours est lue et chargée dans les registres de sortie de la matrice d'état. La PROM de matrice d'état dans le prototype possède une capacité de 32K x 16. Chaque matrice d'état dans la PROM a une capacité de 128 x 16, ce qui fournit un total de matrices à 256 états. Chaque entrée de code d'événement dans une matrice d'état a 32 bits, ce qui fournit un maximum de 64 codes d'événements logiques (une matrice d'état logique a 64 entrées d'événements contenant chacune 32 bits de données de sortie). La Fig.11 est un gabarit de codage général pour chacune des matrices d'état (comme présenté par le programme d'assembleur de PROM de matrice d'état qui a été développé). Il montre les

entrées de code d'événements possibles. Parmi un maximum de 64 entrées de code d'événements, seulement 48 sont nécessaires normalement et codés dans la PROM ECG. Les seize codes d'événements restants sont réservés et les entrées de matrice d'état associée représentée dans la Fig.11 sont désignés par " événement invalide X ". Chaque entrée de code d'événement dans une matrice d'état produit quatre paramètres de sortie à huit bits. Ce sont " Code d'état suivant ", " Code d'état ", " Code d'action " et " Sortie ACK/ECK ".

2.2.4 - Opérations du séquenceur de commande sur les données de sortie de la matrice d'état.

Le séquenceur de commande se connecte sur la sortie de code d'action à huit bits qui est dirigée vers la logique de saut du séquenceur de commande. Le code d'action et la routine d'action résultante détermine quelles opérations sont à réaliser sur les sorties de la PROM de matrice d'état restante et quelles opérations doivent être effectuées pour terminer le reste du cycle de trame de bus RTCN en cours.

La sortie d'état suivante à 8 bits est toujours stockée dans la position de paramètre d'état en cours dans le PCB (dans le module SM) en remplacement du paramètre d'état en cours qui a été utilisé dans l'opération de matrice d'état (voir Fig.4). Dans certains cas, l'état suivant stocké dans le PCB peut être le même que l'état en cours auquel il a été accédé à partir du PCB (c'est à dire que l'événement détecté ne résulte pas en une transition d'état).

La sortie de code d'action peut imposer que la sortie de code d'état soit stockée dans le PCB. Dans ce cas, la routine d'action sera également dirigée pour lire la valeur de l'index du PCB dans la file de répartition d'étape de mots (file de priorités dans

le module SM - voir 1.4.2.2.3) ce qui entraînera le NOS, résidant dans le module IOSF du NIU à réaliser une étape de mots (traitement) sur le PCB.

Si l'état en cours est valide pour l'initialisation de commande de port E/S le code de commande est valide et l'opération de commande de port d'entrée/sortie E/S commence avec une opération de transmission de trame de bus RTCN, la sortie de code d'action commandant que la valeur d'index du PCB soit écrite dans la file de conflit de bus (file de priorité sur le module SM - voir 1.4.2.2.1).

La sortie ACK/ECC de trame à 8 bits n'est jamais utilisée dans le traitement d'événement de commande de port.

2.3 - Traitement d'événement de code d'accusé de réception de trame de réception.

La Fig.12 illustre le traitement d'événement de code d'accusé de réception de trame reçue qui est décrit comme suit.

2.3.1 - Détection d'événement et ensemble de paramètres d'entrée de matrice d'état.

Si le séquenceur de commande transmet une trame de bus RTCN pendant le cycle de trame précédent, il sauvegarde la valeur de l'index PCB associé et valide la réception du code d'accusé de réception de trame à trois bits et le code de correction d'erreurs associé à quatre bits, (ACK/ECC) pendant le cycle de trame en cours. Le séquenceur convertit l'index PCB sauvegardé en l'adresse du segment de commande du PCB. Il lit le paramètre d'état du port en cours du PCB.

2.3.2 - Compression de données d'événement et d'entrée et de sortie de PROM du générateur de code d'événement (ECG) de la matrice d'état.

Le séquenceur établit le code de fonction à deux bits pour l'événement de code d'accusé de réception de trame de réception dans les registres d'événement de la logique de matrice d'état. Il transfère l'ACK/ECC de trame à 7 bits de la FIFO de données de réception au registre d'événement. Le code de fonction établi précédemment, entraîne l'ensemble de données d'événements appropriés dans les registres d'événement hors du bus des données du BIA interne comme représenté sur la Fig.11. Ceci initialise un accès de lecture de la PROM du générateur de code d'événement (ECG) de la logique de matrice d'état. La sortie de la PROM ECG est un code d'événement à six bits qui est dirigée vers les bits 1 à 6 de la PROM de matrice d'état.

La répartition de la PROM ECG, le codage du segment ACK/ECC de trame de réception et la compression de données d'événement ACK/ECC de trame de réception résultante, sont représentés sur la Fig. 13. Comme représenté sur la Fig. 13 le code de fonction dans le format d'entrée, réalise des segments dans la PROM ECG de 1K pour chaque type d'événement. La PROM physique utilisée dans le prototype (8K x 8) a deux fois la capacité requise (4K x 8). L'opération de la PROM ECG comprime les 128 combinaisons de données d'entrée d'événement ACK/ECC de trame de réception en quatre codes d'événements à 6 bits. L'opération et le codage de la PROM ECC fournissent la détection d'erreur de bits unique et la correction et la détection de bits double ou triple associées à l'ECC. Les codes d'ACK (réservés/inutilisés) invalides sont détectés. Les positions de PROM ECG invalides ou inutilisés dans ce segment d'événement sont codés pour produire un code d'événement de sortie "position ECG invalide". Sans la compression de données fournie par

l'étage de la PROM ECG de la logique de matrice d'état, la PROM de matrice d'état aurait du avoir une taille 16 fois supérieure à la taille actuelle (8 mégabits à la place 512 K bits).

5 2.3.3. Entrée et sortie de la PROM de la matrice d'état.

Le code d'état en cours à 8 bits lus du PCB (2.3.1. plus haut) est transféré par le séquenceur aux 8 bits d'ordre supérieur du registre d'état en cours
10 de la logique de matrice d'état avant l'initialisation du fonctionnement de la PROM ECG. La sortie de code d'événement à 6 bits de la PROM ECG est concaténée avec les 6 bits d'ordre inférieur du registre d'état en cours initialisant un accès de lecture de la PROM
15 de matrice d'état. L'entrée de code d'événement à 32 bits spécifiques dans la matrice d'état associée à l'état en cours est lue et chargée dans les registres de sortie de la matrice d'état. La PROM de la matrice d'état dans le prototype a une capacité de 32K x 16.
20 Chaque matrice d'état dans la PROM a une capacité de 128 x 16 ce qui fournit au total 256 matrices d'états. Chaque entrée de code d'événement dans la matrice d'état a 32 bits ce qui fournit un maximum de 64 codes d'événement logique (une matrice d'état logique a 64
25 entrées d'événement contenant chacune 32 bits de données de sortie). La Fig. 11 est un gabarit de codage général pour chacune des matrices d'état (comme présenté par le programme d'assembleur de PROM de matrice d'état qui a été développé). Elle représente
30 des entrées de code d'événement possibles. Sur un maximum de 64 entrées de code d'événement uniquement 48 sont généralement nécessaires et codés dans la PROM ECG. Les 16 codes d'événement restants sont réservés et les entrées de matrice d'état associées comme re-

présenté dans la Fig. 11 sont désignées par "événement invalide X". Chaque entrée de code d'événement, dans la matrice d'état produit quatre paramètres de sortie à 8 bits. Ce sont "code d'état suivant", "code d'état", "code d'action" et "sortie ACK/ECC".

Opération du séquenceur de commande sur les données de sortie de la matrice d'état.

Le séquenceur de commande se connecte sur la sortie de code d'action à 8 bits qui est dirigée vers la logique de saut du séquenceur de commande. Le Code d'action et la routine d'action résultante déterminent quelles opérations sont à effectuer sur les sorties de PROM de matrice d'état restantes et quelles opérations sont à effectuer pour terminer le reste du cycle de trame de bus RTCN en cours. Le jeu de codes d'action et les opérations associés établis pour la configuration de mise en oeuvre sont définis en A.8 des annexes.

La sortie d'état suivante à 8 bits est toujours stockée dans la position de paramètres d'état en cours dans le PCB (dans le module SM) en remplacement du paramètre d'état en cours qui a été utilisé dans l'opération de matrice d'état (voir Fig. 4). Dans certains cas l'état suivant stocké dans le PCB peut être le même que l'état en cours auquel on a accédé à partir du PCB (c'est-à-dire que l'événement détecté ne résulte pas en transition d'état).

La sortie de code d'action peut déterminer que la sortie de code d'état peut également être stockée dans le PCB. Dans ce cas la routine d'action peut également commander l'écriture de la valeur d'index du PCB dans la file de répartition d'étape de travail (file de priorité sur le module SM -voir 1.4.2.2.3.) ce qui entraîne le NOS résidant dans le

module IOSP du NIU à réaliser une étape de travail (traitement) sur le PCB.

La sortie ACK/ECC de ces trames à 8 bits n'est jamais utilisée dans le traitement d'événement de code d'accusé de réception de trame de réception.

2.4. Traitement d'événement de champs de commande de trame de réception.

La Fig. 14 illustre le traitement d'événement de champs de commande de trame de réception décrit comme suit.

2.4.1. Détection d'événement et ensemble de paramètre d'entrée de matrice d'état.

Si un autre noeud entre en conflit et gagne l'accès à la transmission au bus RTCN pendant un cycle de trame donné le séquenceur de commande valide la réception des champs de commande de trame transmis (voir Fig. 7) dans la FIFO de données de réception. Il lit le paramètre LPI de cycle visé de trame reçue de la FIFO de données de réception. Il utilise ce paramètre pour lire le segment de mémoire LPI sur le module SM (voir 1.4.2.2.8). Il vérifie le bit de validation dans l'entrée de mémoire LPI. Si LPI est validé l'on utilise la valeur d'index du PCB fourni dans l'entrée de mémoire LPI pour lire le segment de commande du PCB associé (voir Fig. 4) sur le module SM. Il lit le paramètre d'état de port en cours du PCB.

2.4.2. Compression de données d'événement, d'entrée et de sortie de PROM de générateur de code d'événement (ECG) de matrice d'état.

Le séquenceur établit le code de fonction à 2 bits pour un événement de champ de commande de trame de réception dans les registres d'événement de la logique de matrice d'état. Ceci entraîne le passage du bit d'état vide de la file FIFO d'allocation de cel-

lule d'événement (du module SM) dans les registres d'événement (un bit unique connecté directement à l'entrée du registre d'événement). Il transfère le code de type de trame à 4 bits (FTM) et le code de modification de trame à 4 bits (FTM) de la FIFO de données de réception dans les registres d'événement. Le code de fonction établi précédemment entraîne l'ensemble de données d'événement approprié dans les registres d'événement hors du bus de données de BIA interne comme représenté sur la Fig. 14. Dans ce cas une condition de disponibilité de ressource interne est combinée avec les données d'événement externe. Ceci initialise un accès de lecture de la PROM de générateur de code d'événement (ECG) de la logique de matrice d'état. La sortie de la PROM ECG est un code d'événement à 6 bits qui est dirigée vers les bits 1 à 6 de la PROM de matrice d'état. La répartition de la PROM ECG, le codage d'événement de champ de commande de trame de réception, et la compression de données d'événement de champ de trame de réception résultante sont représentés sur la Fig. 15. Comme représenté sur la Fig. 15 le code de fonction dans le format d'entrée détermine dans la PROM ECG des segments de 1 K pour chaque type d'événement. La PROM physique utilisé dans le prototype (8K x 8) a une capacité égale à deux fois celle nécessaire (4K x 8). L'opération de PROM ECG comprime les 512 combinaisons de données d'entrée d'événement de champ de commande de trame de réception en 22 codes d'événements à 6 bits. L'opération et le codage de la PROM ECG fournit une détection de code de champ de commande de trame invalide. Les positions de PROM ECG invalides ou inutilisées dans ce segment d'événement sont codées pour produire un code d'événement de sortie "ECG invalide". Dans la compression de

données fournies par l'étage de PROM ECG de la logique de matrice d'état, la PROM de matrice d'état aurait une taille 16 fois égale à celle actuelle (8 mégabits au lieu de 512 K bits).

5 2.4.3. Entrée et sortie de la PROM de matrice d'état.

Le code d'état en cours à 8 bits lu du PCB (2.4.1. plus haut) est transféré par le séquenceur aux 8 bits d'ordre supérieur du registre d'état en cours de la logique de matrice d'état avant l'initialisation de l'opération de PROM ECG. La sortie de code d'événement à 6 bits de la PROM ECG est concaténée avec les 6 bits d'ordre inférieur du registre d'état en cours initialisant un accès de lecture de la PROM de la matrice d'état. L'entrée de code d'événement à 10 32 bits spécifiques dans la matrice d'état associée avec l'état en cours est lu et chargé dans les registres de sortie de la matrice d'état. La PROM de matrice d'état dans le prototype a une capacité de 32K x 16. Chaque matrice d'état dans la PROM a une 20 capacité de 128 x 16 ce qui fournit au total 256 matrices d'états. Chaque entrée de code d'événement dans la matrice d'état a 32 bits ce qui fournit un maximum de 64 codes d'événement logiques (une matrice d'état logique a 64 entrées d'événement contenant 25 chacune 32 bits de données de sortie). La Fig. 11 est un gabarit de codage général pour chacune des matrices d'état (comme présenté par le programme d'assembleur de PROM de matrice d'état qui a été développé). Il représente les entrées de code d'événements possibles. 30 Sur un maximum de 64 entrées de code d'événements possibles uniquement 48 sont actuellement nécessaires et codées dans la PROM ECG. Les 16 codes d'événement restants sont réservés et les entrées de matrice d'état associées sont comme représentées sur la Fig.

11 désignées par "événement invalide X". Chaque entrée de code d'événement dans une matrice d'état produit 4 paramètres de sortie à 8 bits. Ceci sont "Code d'état suivant", "code d'état", "code d'action" et "sortie ACK/ECC".

2.4.4. Opération du séquenceur de commande sur les données de sortie de la matrice d'état.

Le séquenceur de commande est connecté sur la sortie de code d'action à 8 bits qui est dirigé vers la logique de saut de séquenceur de commande. Le code d'action et la routine d'action résultante impose les opérations à effectuer sur le reste des sorties de la PROM de matrice d'état et les opérations à effectuer pour achever le reste du cycle de trame de bus RTCN en cours. Le jeu de code d'action et les opérations associées établies pour cette conception d'application sont définies en A8 de l'annexe.

La sortie d'état suivante à 8 bits est toujours stockée dans la position de paramètres d'état en cours dans le PCB (dans le module SM) en remplacement du paramètre d'état en cours qui a été utilisé dans l'opération de matrice d'état (voir la Fig. 4). Dans certains cas l'état suivant stocké dans le PCB peut être le même que l'état en cours auquel on avait accédé à partir du PCB (c'est-à-dire que l'événement détecté ne résulte pas dans une transition d'état).

La sortie de code d'action peut imposer que la sortie de code d'état soit également stockée dans le PCB. Dans ce cas la routine d'action sera également commandée pour écrire la valeur d'index du PCB dans la file de répartition d'étapes de travail (file de priorité sur le module SM -voir 1.4.2.2.3) ce qui entraîne le NOS résidant dans le module IOSF du NIU à réaliser une étape de mots (traitement) sur le PCB.

Le code d'action peut imposer que la sortie

ACK/ECC de trame à 8 bits de la matrice d'état soit sauvegardée par le séquenceur et transmise pendant le cycle de trame de bus de RTCN suivant.

Le code d'action peut également commander le séquenceur pour le transfert des champs de commandes de trames reçus vers un tampon de cellules d'événement. Dans ce cas le séquenceur lit une entrée de la file FIFO d'allocation de cellules d'événement. L'entrée contient l'index d'un tampon de cellules d'événements disponibles (sur le module SM) - voir 1.4.2.2.8). Le séquenceur transfère les champs de commandes de trames reçus de la FIFO de données de réception au tampon de cellule d'événement spécifié. Le séquenceur valide la réception d'une donnée de trame consécutive du bus RTCN à la FIFO de données de réception.

Le code d'action peut commander le séquenceur pour transférer les champs de commandes de trames reçues à un segment de tampons de trames reçus dans le PCB. Dans ce cas, le séquenceur transfère les champs de commandes de trames reçus de la FIFO de données de réception au PCB en utilisant l'index du PCB établi en 2.4.1. ci-dessus. Le séquenceur valide la réception d'une donnée de trame consécutive du bus RTCN à la FIFO de données de réception.

Le code d'action peut commander les séquenceurs pour rejeter des données de champs de commandes de trames dans la FIFO de données de réception et procéder directement à la synchronisation de cycle de trame suivante.

2.5. Traitement d'événement de données de trame de réception.

La Fig. 16 illustre le traitement d'événement de données de trame de réception qui est décrit

comme suit.

2.5.1. Détection d'événement et ensemble de paramètres d'entrées de matrices d'état.

Si la sortie de code d'action de la matrice d'état à partir du traitement d'événement de champs de commande de trame de réception (voir 2.4. ci-dessus) dirige le séquenceur pour recevoir des données de trame, le séquenceur autorise la réception de données de trame de bus RTCN dans la FIFO de données de réception. Le séquenceur valide le générateur de signature (voir 1.4.1.2.3.). Le séquenceur transfert ensuite les données de trames de la FIFO de données de réception vers l'un des trois tampons de données possibles sur le module SM (tampon de file de données (DQ), cellules d'événement, ou segment de tampon de données de trame reçues PCB-- les deux derniers étant destinés à des données de trame de signaux uniquement) comme spécifié par le code d'action. Il lit ensuite le paramètre d'état du port en cours du PCB en utilisant l'index du PCB sauvegardé du traitement d'événement de champs de commande de trame de réception reçu antérieur.

2.5.2. Compression de données d'événement, d'entrée et de sortie de PROM de générateur de code d'événement (ECG) de matrice d'état.

Le séquenceur établit le code de fonction à 2 bits pour l'événement de données de trame de réception dans les registres d'événement de logique de matrice d'état. Ceci entraîne que la sortie "validation de données" du générateur de signature est dirigé vers les registres d'événement (1 bit unique connecté directement aux entrées du registre d'événement). Ceci à son tour initialise un accès de lecture de la PROM du générateur de code d'événement (ECG) de la logique

de matrice d'état. La sortie de la PROM ECG est un code d'événement à 6 bits qui est dirigé vers les bits 1 à 6 de la PROM de matrice d'état. La répartition de la PROM ECG, le codage du segment d'événement de donnée de trame de réception et la compression de données d'événement de données de trame de réception résultantes sont représentées sur la Fig. 17. Comme représenté sur la Fig. 17 le code de fonction dans le format d'entrée réalise dans la PROM ECG des segments de 1K pour chaque type d'événement. La PROM physique utilisée dans le prototype 8K x 8 a une capacité deux fois supérieure à celle nécessaire (4K x 8). L'opération de PROM ECG ne réalise pas de compression de données sur les données d'entrée d'événement de donnée de trame de réception étant donné qu'il n'y a que deux combinaisons de donnée d'entrée. Les positions de PROM ECG invalides ou inutilisées dans ce segment sont codées pour produire un code d'événement de sortie "position ECG invalide".

2.5.3. Entrée et sortie de la PROM de matrice d'état.

Le code d'état en cours à 8 bits lu du PCB (2.5.1. ci-dessus) est transféré par le séquenceur aux 8 bits d'ordre supérieur du registre d'état en cours de la logique de matrice d'état avant initialisation de l'opération de PROM ECG. La sortie de code d'événement à 6 bits de la PROM ECG est concaténée avec les 6 bits d'ordre faible du registre d'état en cours initialisant un accès de lecture de la PROM de matrice d'état. L'entrée de code d'événement à 32 bits spécifiques dans la matrice d'état associé à l'état en cours est lu et chargé dans les registres de sortie de la matrice d'état. La PROM de matrice d'état dans le prototype a une capacité de 32K x 16. Chaque matrice d'état dans la PROM a une capacité de 128 x 16 ce qui

fournit au total 256 matrices d'états. Chaque entrée de code d'événement dans la matrice d'état a 32 bits, ce qui fournit un maximum de 64 codes d'événements logiques (une matrice d'état logique a 64 entrées d'événements contenant chacune 30 bits de données de sortie). La Fig. 11 est un gabarit de codage général pour chacune des matrices d'état (comme présenté par le programme d'assembleur de PROM de matrice d'état qui a été développé).

Elle représente des entrées de code d'événements possibles. Sur un maximum de 64 entrées de code d'événement, uniquement 48 sont nécessaires actuellement et codés dans la PROM ECG. Les seize codes d'événement restants sont réservés et les entrées de matrice d'état associées comme représenté sur la Fig. 11 sont désignées par " événement invalide X ". Chaque entrée de code d'événement dans une matrice d'état produit quatre paramètres de sortie à huit bits. Ceux-ci sont " Code d'état suivant ", " Code d'état ", " Code d'action " et " Sortie ACK/ECC ".

2.5.4 - Opération du séquenceur de commande sur les données de sortie de la matrice d'état.

Le séquenceur de commande est connecté sur la sortie de code d'action à huit bits qui est dirigé vers la logique de saut du séquenceur de commande. Le code d'action et la routine d'action résultante commande quelles opérations doivent être effectuées sur les sorties de la PROM de matrice d'état restante et quelles opérations doivent être effectuées pour terminer le reste du cycle d'état de bus RTCN en cours.

La sortie d'état suivante à 8 bits est toujours stockée dans la position de paramètres d'état en cours dans le PCB (dans le module SM) en remplacement du paramètre d'état en cours qui a été utilisé

dans l'opération de matrice d'état (voir Fig.4). Dans certains cas, l'état suivant stocké dans le PCB peut être le même que l'état en cours auquel on a eu accès à partir du PCB (c'est-à-dire que l'événement détecté ne résulte pas en une transition d'état).

La sortie de code d'action peut commander que la sortie du code d'état soit également stockée dans le PCB. Dans ce cas, la routine d'action va être également commandée pour écrire la valeur d'index du PCB dans la file de répartition d'étape de travail (file de priorité sur le module SM - voir 1.4.2.2.3), ce qui va entraîner le NOS, résidant dans le module IOSP du NIU, à réaliser une étape (de traitement) de travail sur le PCB.

Le code d'action peut commander que la sortie ACK/ECC de la trame à huit bits de la matrice d'état soit conservé par le séquenceur et transmise au cours du cycle de trame du bus RTCN suivant.

SECTION 3 - EXEMPLE DE CODAGE DE MATRICE D'ETAT DE COMMANDE DE PORT DE SIGNAUX D'ECHANGE ET D'OPERATIONS.

3.1 - Exemple de commande de signaux d'échange.

Cette section donne un aperçu du codage logique actuelle de la matrice d'état pour les opérations de commande de port BIA E/S. La commande est " signaux d'échange " qui est définie de la manière suivante.

3.2 - Définition de la commande de signaux d'échange.

3.2.1 - But

Le but de cette commande est de transmettre une trame de signal, d'attendre ensuite et de recevoir une trame de signal en retour. Le mot d'entrée de mémoire spécifié LPI est écrit dans un emplacement de mémoire spécifiée LPI avant la transmission de la trame de signal. Lors de la fin normale de cette com-

mande, il y a transition de port vers l'état des "signaux de contrôle".

3.2.2 - Paramètres d'entrée/sortie.

NOS doit écrire les paramètres d'entrée suivant dans le PCB de port avant de déclencher cette commande :

segment de commande et de contrôle PCB (voir Fig.4)

- * code de commande
- * drapeau de tampon de trame de transmission
- * adresse LPI associée
- * mot d'entrée de mémoire LPI (indiquant normalement l'association état autorisé et port/index PCB)
- * valeur de sortie de temps d'étape de transactions
- * segment de tampon de trame de transmission NOS étape de travail ID et code de priorité PCB
- * mots 1 à 13 pour la trame de signal à transmettre

Le BIA retourne les paramètres de sortie suivant sur le PCB de port en résultat de la fin de l'exécution de la commande normale ou lors de la fin d'une exception :

Segments de commande et de contrôle PCB (voir Fig.4).

- * code d'état de réponse de commande
- * code ACK/ECC de trame reçue
- * segment de tampon de trame de réception PCB
- * toute trame de signal reçu dans les mots 1 à 13

3.2.3 - Opérations.

Cette commande peut être délivrée sur un port soit dans un état d'activation ouvert ou un état de contrôle. La portion de transmission de signal de trame de cette opération de commandes assure de façon
5 similaire la commande de " signal d'envoi" comprenant un nouvel essai sur l'état d'exception ACK/ECC retourné et la préemption de trame de signal reçu. Lors d'une transmission réussie de la trame de signal, au lieu de prendre fin, cette commande assure une at-
10 tente et la réception de toute trame de signal retourné.

Si une trame de signal est reçue, et les données de signal reçu sont bonnes, le BIA retourne un
15 " BON " ACK/ECC et termine l'opération de commande en écrivant l'état de réponse de commande " fin de commande normale" dans le PCB et en écrivant l'index ID/PCB de port, la phase de travail NOS ID est le code de priorité dans la file de réponse de commande de BIA, ce qui provoque la génération d'une requête
20 d'interruption vers le processeur HOTE (PC_AT). Les données de signal reçu sont écrites dans le segment de tampon de trame de réception du PCB. Le port assure la transition à l'état " signaux de commande " .

Si de mauvais signaux de données sont reçus,
25 le BIA retourne un ACK/ECC "RED" et attend de recevoir une retransmission. Il procède de cette manière deux fois seulement. Si lors de la troisième réception du signal de trame, les données de signal reçu sont toujours mauvaises, le BIA retourne un ACK/ECC "RED" et
30 termine l'exécution de la commande avec un état de réponse de commande "données de signal reçu mauvaises" inscrit dans le PCB. La dernière trame de signal reçu sera écrite dans le segment de tampon de trame de réception du PCB.

Si soit un jeton, soit une trame BAL est reçu au lieu d'une trame de signal, le BIA retourne un ACK/ECC "SEQE" et termine l'exécution de la commande avec un état de réponse de commande " exception de
5 protocole" écrit dans le PCB.

La trame de jeton ou de BAL est reçue dans le segment de tampon de réception du PCB.

Si une trame de signal contenant soit un "sens de lecture", un état "PM/FL de lecture " , des
10 "drapeaux de gestion de lecture " , un " drapeau de gestion de test et de réglage " ou " un drapeau de gestion de maintien " il y a réception d'une commande immédiate, le BIA retourne un ACK/ECC "exception LPI" et attend la réception d'une autre trame de signal.

Si une commande immédiate "purge " est reçue, le BIA écrit un état désactivé dans l'emplacement de mémoire LPI associé et termine l'exécution de la commande avec un état de réponse de commande "purge
15 reçue" écrite dans le PCB.

Si cette commande se termine avec tout état autre que "fin de commande normale " , le port assure la transition vers la " fin de l'opération " ou l'état "fin de l'opération/contrôle machine " plutôt que
20 l'état " signaux de contrôle " .

25 3.2.4 - Conditions d'état de réponse applicables.

Les conditions d'état de réponse à une commande applicables à cette commande qui sont délivrées en sortie par la logique de la matrice d'état, sont les suivantes.

- 30 * fin de commande normale (01 hex.)
- * mauvais ACK reçu (02 hex.)
- * délai (03 hex.)
- * signal de préemption reçu (04 hex.)
- * HIO de préemption reçu (05 hex.)

44

- * FRX de préemption reçu (06 hex.)
- * Purge reçue (07 hex.)
- * signal de préemption/mauvaise donnée reçue (08 hex.)
- 5 * exception de protocole (09 hex.)
- * mauvaises données de signal reçu (0A hex.)
- * condition de contrôle de machine pas claire (0C hex.)
- * contrôle de programme de commande BIA non valable (20 hex.)
- 10 * contrôle de programme, de séquence de commande BIA illégal (21 hex.)
- * contrôle de programme port occupé (22 hex.)
- 15 * contrôle machine donnée reçue/attente ACK (30 hex.)
- * contrôle machine donnée reçue/attente de trame CTRL (32 hex.)
- * contrôle machine ACK reçu/attente de trame (34 hex.)
- 20 * contrôle machine trame CTRL reçu/attente données (36 hex.)
- * contrôle machine délai/attente données (38 hex.)
- 25 * contrôle machine commande BIA/attente données (3A hex.)
- * contrôle machine événement non valable (40 hex.)
- * contrôle machine emplacement PROM ECG non valable (50 hex.)
- 30

3.3 - Diagramme de transition d'état de niveau haut.

Les diagrammes de transition d'état de niveau haut font parties de la documentation du codage de logique de la matrice d'état. Ils identifient tous

les états et les matrices d'état associées avec une opération de commande de port d'entrée/sortie de BIA. Ils montrent également les transitions d'état pour les catégories d'événement général. Ils référencent les fichiers de code de source de matrice d'état réels. La Fig.18 fournit une clé pour lire les diagrammes de transition d'état de niveau haut. Trois diagrammes de transition d'état de niveau haut couvrent l'opération de l'exemple "commande de signaux d'échange". La Fig. 19 et le diagramme de transition d'état pour l'"état désactivé" qui est l'un des états de port général/com-
mun dans lequel un port doit se trouver pour accepter une commande de port. La Fig.20 est un diagramme de transition d'état de niveau haut pour l'exemple de commande "signaux d'échange". Il couvre la portion de transmission de trame de signal des opérations de commande. Comme représenté sur cette figure, seulement trois matrices d'état sont nécessaires et uniques pour ces opérations de commande. La Fig.21 est un diagramme de matrice d'état de niveau haut pour le programme commun "trame de signal de réception" qui couvre la portion de trame de réception de signal des opérations de commande de l'exemple "échange signaux". Ainsi qu'on peut le voir à partir de ceci, le codage de logique de matrice d'état permet une forme de sous-programmation. Dans l'exposé qui suit, on va suivre principalement les événements normaux et les transitions d'état associées aux opérations de commande d'exemple comme étant telles qu'elles sont illustrées sur les diagrammes de transition d'état de niveau haut.

3.4 - Etat désactivé.

Une commande de port E/S est normalement appliquée à un port lorsqu'il se trouve soit à l'état

commun "désactivé", ou dans l'état "ouvert" (c'est à dire le paramètre d'état en cours dans le segment de commande et de contrôle du PCB de port indique " désactivé" ou " ouvert "). Comme représenté à la Fig.19, le diagramme de transition d'état de niveau haut pour l'état commun " désactivé", un code d'événement de commande valable fait en sorte que le port assure la transition vers l'état initial associé avec les opérations de commande. La Fig.22 est le code de source d'assembleur actuel pour la matrice d'état pour l'état "désactivé". Dans la description qui va suivre, on va examiner les entrées de code d'événements sélectionnés dans ce code de matrice.

Dans l'état désactivé, le port E/S est normalement en attente d'un événement de commande de BIA pour autoriser le port et déclencher les opérations E/S. L'événement normal dans cet exemple est l'événement de commande " signaux d'échange ". Cette entrée d'événement dans le code de source de la matrice d'état se trouve sur la Fig.22b et elle est indiquée par la référence de texte 4. La sortie d'état suivant est "EXSIGTS_WT_RECAK_1" qui est le premier état dans l'exécution de ces opérations de commande (voir 3.5 ci-après). Cet état est stocké dans le PCB en remplaçant l'état désactivé.

La sortie de code d'état est codée "NONE" indiquant qu'aucun état de réponse de commande ne doit être stocké dans le PCB à cette transition d'état. La sortie ACK Out codée "NULL" comme la sortie de code d'action ne spécifie pas une transition ACK/ECC de trame. La sortie de code d'action, qui est dirigée vers la logique de saut du séquenceur est codée "WR_BUSQ/LPI/INITTO". Ce sous-programme d'action du séquenceur qui est normal pour une opération commandée

qui commence avec une transmission de trame sur le bus RTCN assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivante dans le PCB.

5 2. Ecrire l'index de PCB dans la file de conflit de bus.

3. Ecrire le mot d'entrée LPI spécifié dans le PCB pour l'emplacement de mémoire LPI spécifié dans le PCB. (voir le format de segment de commande et de
10 contrôle du PCB sur les Fig.1 à 4).

4. Initialiser le compteur de fin de PCB.

5. Retour aux services de ligne de temps au point normal à partir d'un événement de commande de BIA.

15 Comme représenté dans le code de source de la matrice d'état à la Fig.22, les événements de réception ACK/ECC (référence de texte 1), les événements de réception de champ de commande de trame (référence de texte 2), événement non valable (c'est à dire code
20 d'événement non usuel/de réserve) (référence de texte 3), l'événement de réception de données de trame (référence de texte 7) et emplacement ECG non valable (référence de texte 8) sont sous des conditions de contrôle de machine. Ils sont tous codés avec une
25 sortie d'état suivant de "DISABLED _ MACH _ CHK" qui fait en sorte que le port assure la transition vers l'état de contrôle de machine désactivée. La sortie de code d'état pour chacun des événements est codée pour indiquer la condition de contrôle spécifique de la
30 machine. Les sorties de code d'action pour ces événements sont codées "LOG _ ST/PCBI". Ce sous-programme d'action de séquenceur assure les actions discrètes suivantes :

1. Ecriture de la sortie d'état suivant dans

le PCB.

2. Ecriture de la sortie d'état dans le tampon d'enregistrement de contrôle de machine (sur le module SM).

5 3. Ecriture de l'index PCB dans le tampon d'enregistrement de contrôle de machine.

4. Retour au service de ligne de temps basé sur le code de sortie de ligne de temps préservé avant de passer à la matrice d'état pour la résolution
10 d'événement.

On notera que les événements de trame de réception ACK/ECC, de champs de commande de trame de réception et de données de trame de réception sont des conditions de contrôle de machine, car dans l'état
15 désactivé, le port n'assure pas les opérations E/S et son index PCB n'est entré dans aucune mémoire LPI autorisée (c'est à dire une mémoire à association d'adresse de trame logique). La seule façon de pouvoir associer ces événements à ce port/PCB est le résultat
20 d'une erreur de circuiterie.

Le code de source de matrice d'état de la Fig.22 illustre certaines conditions d'exception de commande.

L'événement "XMIT _ DAT _ CMD" (référence de
25 texte 5) est une séquence de commande illégale. Le port doit être soit à l'état "OUVERT" ou dans l'un des états de "CONTROLE" pour que cette commande puisse être acceptée. L'événement "CMD NON VALABLE" (référence de texte 6) est un code de commande non utilisé/réservé. Dans les deux cas, la sortie d'état sui-
30 vant est codée "DESACTIVE" ce qui fait en sorte que le port reste à l'état désactivé. Les sorties de code d'état sont codées " PC _ ILLEGAL _ CMDSEQ " et " PC _ INVALID _ CMD" respectivement. Les sorties de code

d'action sont codées " WR _ STAT/CRQ/CLTO ". Ce sous-programme d'action de séquenceur assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivant dans le PCB.
2. Effacer le compteur d'attente de PCB.
3. Ecrire la sortie de code d'état dans le PCB.
4. Ecrire l'index de PCB dans la file de distribution de phase de travail.
5. Retour aux services de lignes de temps au point normal à partir de l'événement de commande de BIA.

3.5 - Attente de signaux d'échange à l'état d'accusé de première réception de trame (EXSIGS_WT_RECACK_1).

Comme représenté sur le diagramme de transition d'état de niveau haut de la Fig.20, ceci est l'état initial dans l'exécution d'opérations de commande de signaux d'échange. Dans cet état, (voir les actions de transition d'état précédentes au paragraphe 3.4 ci-dessus) le port attend que le séquenceur transmette la trame de signal et le retour d'un événement de trame de réception ACK/ECC (événement normal dans cet état). La Fig.23 et le code de source PROM de matrice d'état pour cette matrice d'état. Dans la description qui va suivre, on va examiner les entrées de code d'événement sélectionnées dans ce code de matrice.

L'événement normal pour cet état est "GOOD_ACK" (un événement de trame de réception ACK/ECC) qui est représenté dans le code de source de matrice d'état à la Fig.23 (référence de texte 1). La sortie d'état suivant pour cette transition d'état normal est codée "RSIG_WAIT_SIG_1", qui est le premier

état dans les opérations communes de réception de signaux. La sortie de code d'état est codée "NONE" indiquant qu'aucun état de réponse de commande ne doit être stocké dans le PCB à cette transition d'état. La sortie ACK Out codée "NULL" comme la sortie de code d'action ne spécifie pas une transmission de trame ACK/ECC. La sortie de code d'action qui est dirigée vers la logique de saut du séquenceur est codée "WR_ACK_TO_PCB". Ce sous-programme d'action du séquenceur qui est normal pour un événement de réception ACK/ECC qui ne termine pas les opérations de commande de port assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivante dans le PCB.
2. Ecrire ACK/ECC reçu dans le PCB.
3. Retourner au service de ligne de temps au début de conflit de bus de ce cycle de trame.

Les procédures de commande de matrice d'état pour ces opérations de commande vont reessayer des transitions de trame deux fois si soit un événement ACK/ECC de réception de trame " RED_ACK " ou " NULL_ACK " se produit. Le codage pour ceci est illustré par le codage pour l'événement "NULL_ACK" sur la Fig.23 (référence de texte 2). La sortie d'état suivant est codée "EXIGS_WT_RECACK_2". Cet état est similaire à l'état en cours à l'exception du fait qu'il constitue la seconde tentative de transmission (c'est à dire le premier essai réitéré). La sortie de code d'action est codée " WR_BUSQ". Ce sous-programme d'action de séquenceur qui est normal pour un essai réitéré de transmission de trame assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivante dans le

PCB.

2. Ecrire l'index de PCB dans la file
conflit de bus.
3. Retourner au service de ligne de temps au
début de conflit de bus, ce cycle de
trame.

La référence de texte 3 sur la Fig.23 illustre le codage pour certains événements de champs de commande de trame de réception. L'événement est "APERIODIC_BAL". La sortie d'état suivant est codée "EXSIGSWT_RECACK_1) ce qui entraîne que le port reste dans le même état (pas de transition d'état). La sortie ACK/ECC Out est codée "SEQE" (erreur de séquence). La sortie de code d'action est codée "CLR_BAL/SEND_ACK". Ce sous-programme d'action de séquenceur qui est normal pour une réception de trame BAL rejetée assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivant dans le
PCB.
2. Conserver et envoyer la sortie ACK/ECC au
cours du temps ACK de cycle de trame
suivant.
3. Abandonner les données de champ de commande de trame BAL reçues.
4. Retourner aux services de ligne de temps
et abandonner les données de paquets qui
suivent. Envoyer la sortie ACK/ECC au
temps ACK de cycle de trame suivant.

La référence de texte 4, sur la Fig.23, illustre le codage d'événements de commande BIA. L'événement spécifique des "DISABLE_CMD". La sortie d'état suivant est codée "OPEND" qui est un état de fin commun pour les ports qui ont terminé les opérations

d'exécution de commande. La sortie de code d'état est codée "PC_PORT_BUSY". La sortie de code d'action est codée "WR_PG/ST/TO/CRQ". Ce sous-programme d'action de séquenceur qui est normal pour terminer des opérations de commande qui ont un index PCB dans la file de conflit de bus assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivant dans le PCB.
2. Ecrire la sortie de code d'état dans le PCB.
3. Incrémenter le compteur de purge de file conflit de bus dans le PCB. (Ceci fait en sorte que les entrées de file de conflit de bus pour ce port sont purgées au cours de la période de traitement de conflit de trame de bus suivante par le séquenceur.
4. Effacer le compteur d'attente dans le PCB.
5. Ecrire le PCB du port dans la file de distribution de phase de travail.
6. Retourner aux services de lignes de temps basées sur le code de sortie de lignes de temps préservées avant de passer à la matrice d'état pour la résolution d'événement.

Comme représenté dans le code de source de la matrice d'état à la Fig.23, un événement non valable (c'est à dire un code d'événement non utilisé/ de réserve) (référence de texte 5), événement de données de trame reçue, (référence de texte 6) et emplacement ECG non valable (référence de texte 7) sont toutes des conditions de contrôle de machine. Elles sont toutes codées avec une sortie d'état suivant de " OPEND_MACH_CHK" qui oblige le port à effectuer une transition

vers l'état de contrôle de machine OPEND. La sortie de code d'état pour chacun de ces événements est codée pour indiquer la condition de contrôle de machine spécifique. Les sorties de code d'action pour ces événements sont codées " WR_PG/ST/TO/CRQ". Ce sous-programme d'action de séquenceur assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivant dans le PCB.
2. Ecrire la sortie d'état dans le PCB.
3. Incrémenter le compteur de purge de file de conflit de bus dans le PCB.
4. Effacer le compteur d'attente dans le PCB.
5. Ecrire l'index PCB de port dans la file de distribution de phase de travail.
6. Retour aux services de lignes de temps basées sur le code de sortie de lignes de temps conservé avant de passer à la matrice d'état pour la résolution d'événement.

On notera que les événements de données de trame de réception sont des conditions de contrôle de machine, car ils peuvent se produire seulement lorsque le port a reçu un événement de champs, de commande de trame de réception précédent et assure la transition à un état en cours en attendant spécifiquement un événement de données de trame de réception. La seule façon de pouvoir associer ces événements à un port dans un état attendant un événement de réception de trame ACK et le résultat d'une erreur de circuiterie.

3.6 - Attente de signal de réception sur un état de commande de trame de premier signal de réception (RSIG_WT_SIG_1).

Comme représenté dans le diagramme de transition d'état de niveau haut, de la Fig.21, ceci est l'état initial dans l'exécution du sous-programme de matrice d'état de signal de réception commun qui commande la portion de signal de réception des opérations de commande de signaux d'échange. Dans cet état (voir actions de transition d'état précédentes dans 3.5 plus haut), le port attend la réception des champs de commande d'une trame de signal. La Fig.24 est le code de source PROM de matrice d'état pour cette matrice d'état. Dans ce qui va suivre, on va examiner les entrées de code d'événement sélectionnées dans ce code de matrice.

L'événement normal pour cet état est l'évènement de champs de commande de trame de signal; un tel événement est représenté dans le code de source de matrice d'état à la Fig.24 (référence de texte 2). La sortie d'état suivant pour cette transition d'état normal est codée "RSIG_WT_SIGDATA_1", qui est la transition d'état normal et le second état dans les opérations de signaux de réception communs. La sortie de code d'état est codée "NONE" indiquant qu'aucun état de réponse de commande ne doit être stocké dans le PCB à cette transition d'état. La sortie ACK Out est codée "NULL" car la sortie de code d'action ne spécifie pas une transition de trame ACK/ECC. La sortie de code d'action qui est dirigée vers la logique de saut de séquenceur est codée "WR_FRCTRL". Ce sous-programme d'action de séquenceur qui est normal pour un événement de champ de commande de trame de réception assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivant dans le PCB.
2. Ecrire les données de champ de commande

de trame de signal reçu dans le PCB.

3. Retourner aux services de lignes de temps et aux données de trame de signal suivant dans le PCB (recevoir le segment de tampon de données de trame du PCB).

5

Comme représenté dans le code de source de la matrice d'état à la Fig.24, les événements de réception ACK/ECC (référence de texte 1), l'événement non valable (c'est à dire code d'événement non utilisé/ de réserve) (référence de texte 5), l'événement de réception de données de trame (référence de texte 6) et emplacement ECG non valable (référence de texte 7) sont toutes des conditions de contrôle de machine. Elles sont toutes codées avec une sortie d'état suivant de "OPEND_MACH_CHK" ce qui entraîne que le port assure la transition vers l'état de contrôle de machine OPEND. La sortie de code d'état pour chacun de ces événements est codée pour indiquer la condition de contrôle spécifique de la machine. Les sorties de code d'action pour ces événements sont codées "WE_ST/TO/CRQ". Ce sous-programme d'action de séquenceur assure les actions discrètes suivantes :

10

15

20

25

30

1. Ecriture de la sortie d'état suivant dans le PCB.
2. Ecriture de la sortie d'état dans le PCB.
3. Effacement du compteur de délai dans le PCB.
4. Ecriture de l'index de PCB du port dans la file de distribution de phase de travail.
5. Retour aux services de lignes de temps basé sur le code de sortie de lignes de temps conservé avant de passer à la matrice d'état pour la résolution d'événement.

On notera que la trame de réception ACK/ECC, les champs de commande de trame de réception, et les événements de données de trame de réception sont des conditions de contrôle de machine car dans l'état désactivé, le port n'assure pas les opérations E/S et son index de PCB n'est entré dans aucune mémoire LPI autorisée (c'est à dire une mémoire d'association d'adresse de trame logique). La seule manière dont ces événements peuvent être associées à ce port/PCB est un résultat d'une erreur de circuiterie.

Le port reste dans un état occupé tout en exécutant la commande de signaux d'échange. Aucune nouvelle de commande de BIA ne doit être délivrée à ce port jusqu'à ce qu'il passe à l'état OPEND.

Le code de source de matrice d'état de la Fig.24 illustre certaines conditions d'exception de commande. Un événement "XMIT_SIG_CMD" (référence de texte 3) et un événement "INVALID_CMD" (référence de texte 4) (un code de commande non utilisé/réservé). Dans les deux cas, la sortie d'état suivant est codée "OPEND" ce qui entraîne que le port termine les opérations de commande. Les sorties de code d'état sont codées "PC_PORT_BUSY" dans les deux cas. Les sorties de code d'action sont codées "WR_STAT/CRQ/CLTO". Ce sous-programme d'action du séquenceur assure les actions discrètes suivantes :

1. Ecriture de la sortie d'état suivant dans le PCB.
2. Effacement du compteur de délai du PCB.
3. Ecriture de la sortie de code d'état dans le PCB.
4. Ecriture de l'index PCB du port dans la file de distribution de phase de travail.

5. Retour aux services de lignes de temps au point normal à partir de l'événement de commande BIA.

5 3.7 - Réception du signal attente sur un premier état de données (RSIG_WT_SIGDATA_1) de premier signal de réception.

Comme représenté dans le diagramme de transition d'état de niveau haut à la Fig.21, ceci est le second état dans la transition exécution/état normale du sous-programme de matrice d'état du signal de réception commun qui commande la portion de signal de réception des opérations de commande de signaux d'échange. Dans cet état (voir action de transition d'état précédente dans 3.6 ci-dessus) le port est en attente de la réception de données de signal de trame. La Fig.25 et le code de source PROM de matrice d'état pour cette matrice d'état. Dans la description qui va suivre, on va examiner des entrées de code d'événements choisis dans ce code de matrice.

20 L'événement normal pour cet état est un événement "GOOD_DATA" comme représenté dans le code de source de la matrice d'état de la Fig.25 (référence de texte 5). La sortie d'état suivante pour cette transition d'état normale est codée "MONSIGS_WT_SIGFR",
25 qui est la transition d'état normale et la fin normale des opérations de commande. La sortie de code d'état est codée "NORMAL_CMD_END" indiquant la fin des opérations de commande sans aucune condition d'exception. La sortie ACK Out est codée "GOOD" et sera transmise
30 au cours du cycle de trame suivant. La sortie de code d'action qui est dirigée vers la logique de saut du séquenceur est codée "WR_ST/CRQ/SEND_ACK". Cette routine d'action de séquenceur qui est normale pour une opération commandée qui se termine avec la

réception des données de trame correcte assure les actions discrètes suivantes :

1. Ecrire la sortie d'état suivante dans le PCB.
- 5 2. Ecrire la sortie de code d'état dans le PCB.
3. Conserver la sortie la sortie ACK/ECC Out et envoyer le cycle trame suivant.
4. Effacer le compteur de délai dans le PCB.
- 10 5. Ecrire l'index PCB de port dans la file de distribution de phase de travail.
6. Retourner aux services de lignes de temps et envoyer la sortie ACK/ECC préservée au cours du cycle de trame suivant.

15 Comme représenté dans le code de source de matrice d'état à la Fig.25, les événements réception ACK/ECC (référence de texte 1), événement de réception de champ de commande de trame (référence de texte 2), événement non valable (code d'événement non utilisé/

20 réservé) (référence de texte 3), événement de commande BIA (référence de texte 4), et événement d'emplacement ECG non valable (référence de texte 7) sont tous des conditions de contrôle de machine. Ils sont tous codés avec une sortie d'état suivant de "OPEND_MACH_CHK" ce

25 qui entraîne que le port assure la transition vers l'état de contrôle de machine OPEND. La sortie de code d'état pour chacun de ces événements est codée pour indiquer la condition de contrôle de machine particulière. Les sorties de code d'action pour ces

30 événements sont codées "WR_ST/TO/CRQ". Ce sous-programme d'action de séquenceur assure les actions discrètes suivantes.

1. Ecrire la sortie d'état suivant dans le PCB.

2. Ecrire la sortie d'état dans le PCB.
3. Effacer le compteur de délai dans le PCB.
4. Ecrire l'index PCB de port dans la file de distribution de phase de travail.

5 5. Retourner au service de ligne de temps sur la base du code de sortie des lignes de temps préservé avant de passer à la résolution d'événement pour la matrice d'état.

10 On notera que les événements de réception de trame ACK/ECC, les événements de champs de commande de trame de réception, et les événements de commande BIA sont des conditions de contrôle de machine car l'événement précédent qui a provoqué la transition d'état vers l'état en cours était un événement de
15 champ de commande de signal de trame de réception et aucun des événements ci-dessus ne peut se produire avant un événement de donnée de trame de réception. La seule façon dont ces événements peuvent se produire dans l'état en cours résulte d'une erreur de
20 circuiterie.

 La portion de trame de signal de réception de commande des signaux d'échange assure deux tentatives réitérées si de mauvaises données de trame de signal sont reçues comme représenté dans le code de
25 source de matrice d'état à la Fig. 25 (référence de texte 6). La sortie d'état suivant d'entrée d'évènement "REC_DATA_BAD" est codée "RSIG_WAIT_SIG_2". Ce nouvel état (un code de matrice d'état associé) est similaire a RISG_WAIT_SIG_1" dans 3.6 ci-dessus. Cette
30 transition d'état (voir le diagramme de transition d'état de niveau haut à la Fig. 21) fait en sorte que la procédure attende un événement de champ de commande de trame de signal de réception associé à la retransmission de la trame de signal. La sortie ACK/ECC est codée "RED" (Erreur de répétition détectée) qui est

transmise au cycle de trame suivant. La sortie de code d'action est codée "SEND_ACK". Ce sous-programme d'action de séquenceur assure les actions discrètes suivantes :

- 5 1. Ecrire la sortie d'état suivante dans le PCB.
2. Préserver la sortie ACK/ECC et envoyer le temps ACK de cycle de trame suivant.
3. Retourner aux services de ligne de temps
10 ACK/ECC suivant conservé.

SECTION 4 - Résumé des résultats d'application décrite 4.1. Efficacité d'utilisation de ressource.

La Fig. 26 résume les ressources de logique de matrice d'état requises et utilisées pour mettre en
15 oeuvre l'application RTCN NIU BIA. Elle indique les ressources disponibles, celles utilisées, et celles qui restent en réserve. L'application a nécessité la mise en oeuvre de 18 procédures de commande de port E/S de BIA, un ensemble de procédure de commande pour
20 le port spécial (BSG) de générateur de synchronisation et un jeu de procédures de commande pour les LPI physiques dérivés (PDL) (c'est-à-dire port de contrôle de neud physique). Cependant la mise en oeuvre a utilisé
25 seulement 73 (30%) parmi les 256 matrices d'état disponibles dans la PROM de matrices d'état.

L'invention résultante fournit un processeur de données en temps réel, commandé par évènement perfectionné qui est plus souple en ce qu'il permet une reconfiguration de combinaisons d'évènement et de ré-
30 ponses désirées, assure une résolution plus rapide des transitions d'état de processus de commande et d'action de commande requise et il est capable de stocker un large répertoire de combinaisons d'évènements et de réponses désirées, avec une attribution de mémoire

plus faible que ce qui a été nécessaire dans l'état de la technique. L'invention peut être appliquée au traitement de données en temps réel commandé par événement dans la commande de processus industriel, la commutation dans les communications, le traitement de données distribué, la commande de moteurs à combustion interne, la détection et la prévention de collisions ainsi que dans de nombreux autres domaines.

Bien qu'un mode de réalisation particulier de l'invention ait été décrit, les spécialistes comprendront que des modifications peuvent être apportées à ce mode de réalisation particulier sans sortir de l'esprit et du cadre de l'invention.

REVENDEICATIONS

1. Processeur de données en temps réel, commandé par événement ayant une entrée connectée à une source de signaux d'événement et une sortie de
5 commande connectée à des éléments à commander, caractérisé en ce qu'il comprend une mémoire de matrice d'état ayant une première entrée connectée à ladite entrée d'événement et une seconde entrée connectée pour recevoir une indication d'état suivant et
10 ayant une sortie pour stocker plusieurs matrices d'état, chaque matrice d'état étant choisie par une valeur de l'indication d'état suivant sur ladite seconde entrée correspondant à un état de commande et organisée en rangées et en colonnes, chaque rangée
15 correspondant à un événement devant être détecté et chaque rangée étant divisée en éléments d'une première colonne pour l'état suivant et d'une seconde colonne pour l'action, un séquenceur de commande ayant une entrée connectée à la sortie de ladite mémoire de
20 matrice d'état pour recevoir des données de rangées à partir d'une matrice sélectionnées dans ladite mémoire de matrice, en réponse à ladite mémoire de matrice d'état ayant reçu une information d'événement et d'état suivant sur ses première et seconde entrées,
25 ledit séquenceur de commande accédant en réponse à celle-ci à une séquence d'instructions exécutables, une unité arithmétique/logique ayant une entrée connectée à une sortie dudit séquenceur de commande, pour recevoir ladite séquence d'instructions
30 exécutables en vue de l'exécution dans celle-ci, lesdites instructions comprenant une nouvelle valeur pour l'état suivant du système qui est délivrée en sortie à ladite seconde entrée de ladite mémoire de matrice d'état, autorisant ainsi ladite mémoire de

matrice d'état à répondre à la production d'un événement suivant.

5 2. Appareil suivant la revendication 1, caractérisé en ce qu'il comporte en outre un générateur de code d'événement connecté entre ladite entrée d'événement et ladite première entrée de ladite mémoire de matrice d'état pour compresser des données d'événement devant être appliquées à ladite première entrée de ladite mémoire de matrice d'état.

10 3. Appareil suivant la revendication 1, caractérisé en ce que le séquenceur établit et détecte des événements qui doivent être présentés à la logique de matrice d'état en vue de la résolution par rapport à l'état de traitement de commande en cours.

15 4. Appareil suivant la revendication 1, caractérisé en ce que les instructions de séquenceur et les possibilités d'adressage de l'opérante de source et de destination transfèrent des données d'état en cours et d'événements aux entrées de la
20 logique de la matrice d'état et déclenchent des opérations logiques de matrice d'état.

 5. Appareil suivant la revendication 1, caractérisé en ce que le séquenceur peut combiner (concaténer) des conditions d'état interne (telles que
25 la disponibilité de ressource/tampon) avec des données d'événement pour permettre à la logique de matrice d'état d'assurer une résolution vis à vis à la fois de l'état de processus de commande en cours, l'application de "événement de commande de trame de réception"
30 dans RTCN BIA étant un exemple.

 6. Appareil suivant la revendication 1, caractérisé en ce que la sortie "code action" de la logique des matrices d'état est câblée directement à la logique de saut du séquenceur où des instructions

spéciales "branchement sur code d'action de matrice d'état" agissent sur celui-ci.

5 7. Appareil suivant la revendication 1, caractérisé en ce que "le code d'action" spécifie la réponse du séquenceur de commande requise et provoque en réponse à l'événement la continuation ou la fin du cycle de processus de commande.

10 8. Appareil suivant la revendication 1, caractérisé en ce que les instructions de séquenceur et les fonctions d'adressage d'opérante de destination et de source associée transfèrent l'état suivant de la matrice d'état, le code d'état, et des sorties de code d'accusé de trame à des destinations requises telles que des blocs de commande de port, de transmission de
15 données FIFO ou E/S (dans la mémoire de commande SM).

20 9. Appareil suivant la revendication 1, caractérisé en ce que l'établissement du "code de fonction" en tant que partie du processus d'entrée de la logique de la matrice d'état déclenche différents ensembles de données d'événements (et des données de conditions d'état internes) (masquant) en dehors du bus de données interne dans les registres d'entrée d'événement de logique de matrice d'état.

25 10. Appareil suivant la revendication 1, caractérisé en ce que la logique de matrice d'état est présentée avec des données de matrice d'événement (et/ou des données de condition d'état interne) un code de fonction et des entrées de données d'état de processus de commande en cours.

30 11. Appareil suivant la revendication 1, caractérisé en ce que le code de fonction commande différents ensembles de données d'événement et le reformatage en dehors du bus de données internes.

12. Appareil suivant la revendication 1,

caractérisé en ce que le code de fonction déclenche l'énergie pour les PROMs de la logique de la matrice d'état et la dernière sortie de données provenant de la PROM de la matrice d'état désactive l'énergie, économisant ainsi la quantité d'énergie utilisée par la logique de matrice d'état.

13. Appareil suivant la revendication 1, caractérisé en ce que la logique de la matrice d'état est réalisée par deux opérations PROM (série) en cascade exécutées et commandées en tant qu'opération unique.

14. Appareil suivant la revendication 1, caractérisé en ce que la PROM du générateur de code d'événement (ECG) accepte le code de fonction et les entrées de données d'événement de rangées, assure la compression de données sur les données d'événement et produit une sortie de code d'événement, ladite compression de données dans l'exemple d'application RTCN BIA réduisant les exigences de stockage de la logique de matrice d'état (PROM) d'un facteur 16.

15. Appareil suivant la revendication 1, caractérisé en ce que la PROM ECG assure également des opérations de détection d'erreurs et de corrections sur toutes données d'événement de rangées qui comprennent un ECG, les "événements ACK/ECG de trame de réception" dans l'application RTCN BIA étant un exemple de ceci.

16. Appareil suivant la revendication 1, caractérisé en ce que la PROM de la matrice d'état accepte un code d'événement (provenant de la PROM ECG) et des données d'état de processus courant combinés en tant qu'entrée unique.

17. Appareil suivant la revendication 1, caractérisé en ce que la PROM de la matrice d'état

produit une sortie de "rangée de matrice" correspondant à l'état en cours et à un événement détecté, cette sortie de "rangée de matrice" contenant des sorties de données d'état suivant, un code d'action, d'état de processus et un code d'accusé de trame (c'est-à-dire code de réponse).

18. Appareil suivant la revendication 1, caractérisé en ce que la sortie de code d'action est câblée directement sur la logique de saut du séquenceur.

19. Processeur de données en temps réel, commandé par événement ayant une entrée connectée à une source de signaux d'événement et une sortie de commande connectée à des éléments à commander, caractérisé en ce qu'il comprend une mémoire de matrice d'état ayant une première entrée connectée à ladite entrée d'événement et une seconde entrée connectée à un registre d'état en cours et ayant une sortie pour stocker plusieurs matrices d'état, chacune desdites matrices d'état correspondant à un état de commande et étant organisées, en rangées et en colonnes, chaque rangée correspondant à un événement devant être détecté et chaque rangée étant divisée en éléments d'une première colonne pour l'état suivant et d'une seconde colonne pour l'état, une troisième colonne pour l'action et une quatrième colonne pour l'accusé de sortie, un séquenceur de commande ayant une entrée connectée à la sortie de ladite mémoire de matrice d'état pour recevoir des données de rangées provenant d'une matrice sélectionnée dans ladite mémoire de matrice en réponse à ladite mémoire de matrice d'état, ayant une information d'événement reçu et d'état en cours sur ses première et seconde entrées, ledit séquenceur de commande accédant en réponse à celles-ci

à une séquence instructions exécutables, une unité arithmétique/logique ayant une entrée connectée à une sortie dudit séquenceur de commande pour recevoir ladite séquence d'instructions exécutables en vue de leur exécution dans celle-ci, et parmi lesdites instructions une nouvelle valeur étant présente pour l'état en cours du système qui est écrit dans ledit registre d'état en cours, ladite nouvelle valeur stockée dans ledit registre d'état en cours permettant à la mémoire de matrice d'état de répondre à la production d'un événement suivant.

20. Appareil suivant la revendication 19, caractérisé en ce qu'il comporte en outre un générateur de code d'événement connecté entre ladite entrée d'événement et ladite première entrée de ladite mémoire de matrice d'état pour compresser des données d'événement devant être appliquées à ladite première entrée de ladite mémoire de matrice d'état.

FIGURE 1.

CERTAINES CONFIGURATIONS D'UNITE RTCN TIREES DU BIA
ET D'AUTRES MODULES FONCTIONNELLS DE CIRCUITS COURANTS

LE MODULE BIA CONTIENT UN SEQUECEUR DE COMMANDE EN TEMPS REEL INCORPORANT
UNE LOGIQUE DE MATRICE D'ETAT

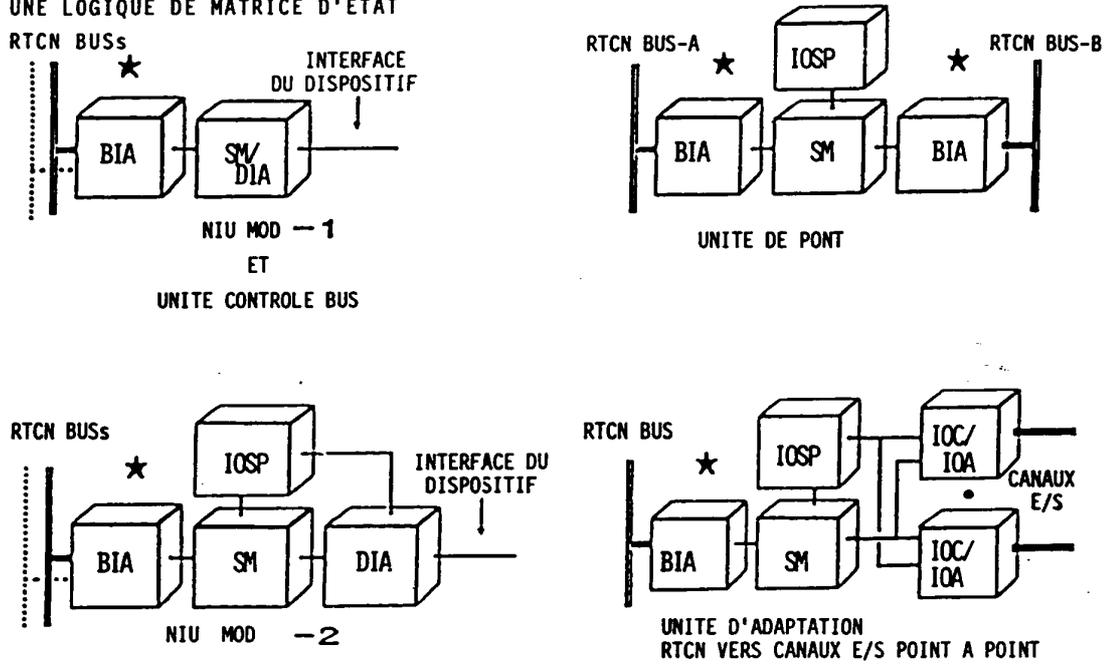


FIGURE 2.
ADAPTATION D'INTERFACE DE BUS (BIA)
SCHEMA BLOC FONCTIONNEL DU MODULE

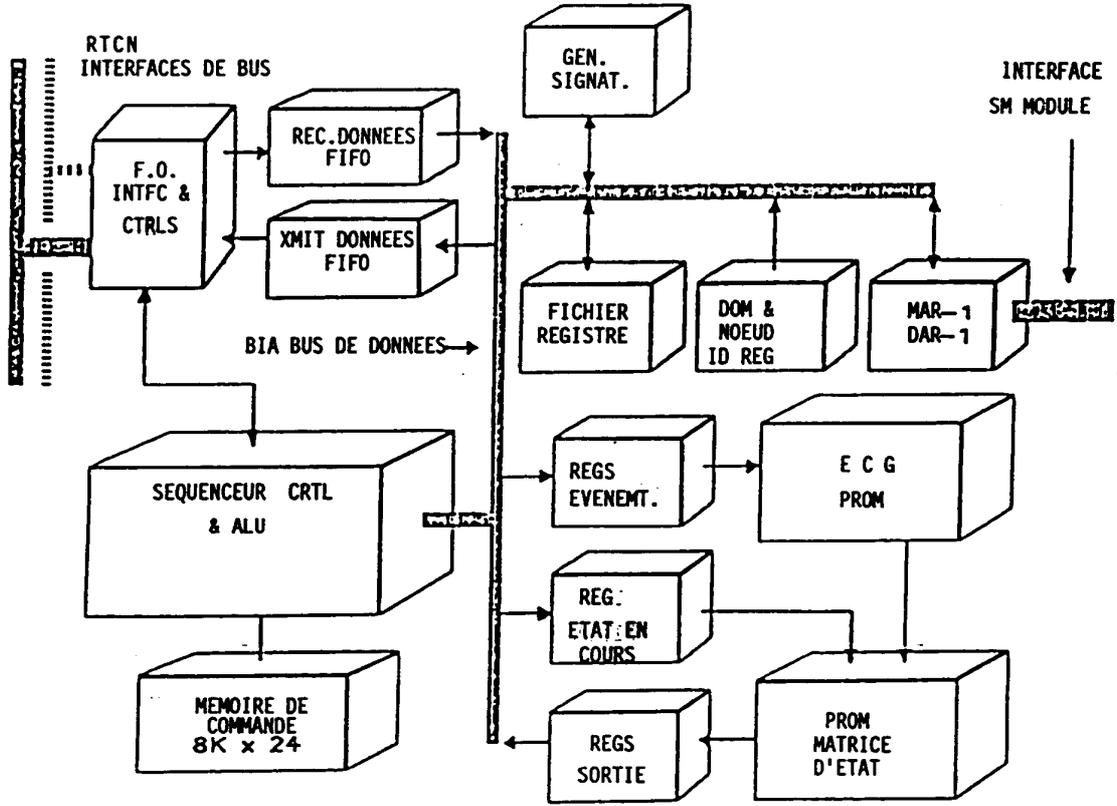
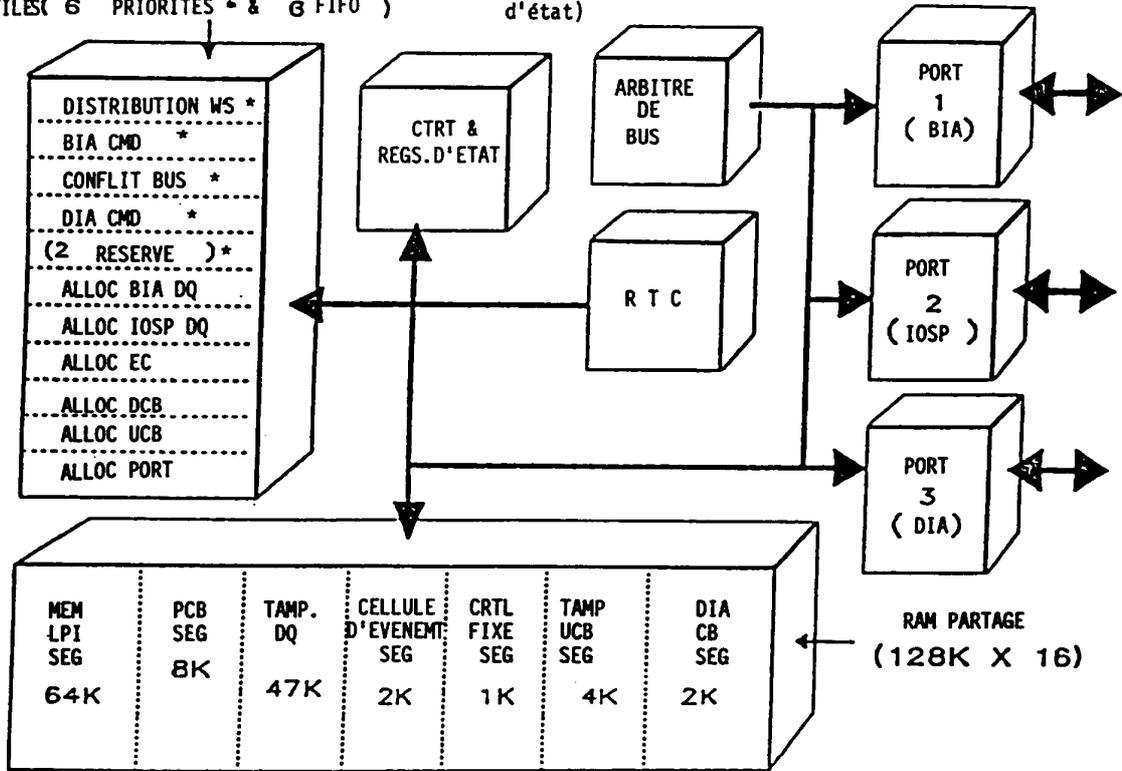


FIGURE 3.

SCHEMA BLOC FONCTIONNEL DE LA MEMOIRE PARTAGEE (SM)
 (SM contient des éléments de commande auxiliaires associés à des opérations de matrice d'état)

CONTROLEUR DE FILE ET FILES (6 PRIORITES * & 6 FIFO)



4/24

FIGURE 4.

FORMAT DE SEGMENT ET DE COMMANDE PCB

MOT N°	* MOT INDEX (BIN.)	PARAMETRES DE MOT / POSITIONS DE BIT															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	000000	* VECTEUR D'ETAT EN COURS **								- XX - (RESERVE POUR UTILISATION BIA)							
2	000001	-XX- (RESERVE POUR UTILISATION BIA)								CODE DE PURGE DE FILE DE CONFLIT DE ** BUS							
3	000010	COMPTEUR DE TEMPS ECOULE **															
4	000011	(RESERVE POUR UTILISATION BIA)															
5	000100	RESERVE POUR UTILISATION NOS								-0-		CODE BIA CMD *					
6	000101	(RESERVE POUR UTILISATION NOS)														T B	
7	000110	CODE D'ETAT DE REPONSE DE COMMANDE **								-XX-							
8	000111	ADRESSE LPI ASSOCIEE ** -															
9	001000	MOT D'ENTREE DE MEMOIRE LPI **															
10	001001	VALEUR DE TEMPS ECOULE D'ETAPE DE TRANSACTION															
11	001010	-0-								CODE ACK/ECC DE TRAME RECUE**							
12	001011	ADRESSE DE TAMPON DQ															
13	001100	COMPTAGE DE SOUS-PAQUETS DE DONNEES D'ENVOI/DE RECEPTION															
14	001101	REPONSE/APPEL DE TACHE ID								REPONSE/APPEL DE PRIORITE							
15	001110	(RESERVE POUR UTILISATION NOS)															
16	001111	(RESERVE POUR UTILISATION NOS)															

5/24

FIGURE 5.

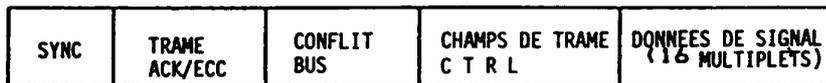
TRAME ZERO



TRAME DE JETON



TRAME DE SIGNAL



TRAME BAL



RECEPTION
EVENEMENT
ACK/ECC*

BIA
COMMANDE
EVENEMENT
*

RECEPTION
EVENEMENT
FR CTRL *

RECEPTION
EVENEMENT
DONNEES
FR *

* QUATRE POINTS DANS UN CYCLE DE TRAME, ETAT AUQUEL DES OPERATIONS LOGIQUES DE MATRICE D'ETAT SONT EFFECTUEES ET LES TYPES D'EVENEMENT ASSOCIES

FIGURE 5 TYPES DE TRAME DE BUS RTCN ET OPERATIONS DE MATRICE D'ETAT DE CYCLE DE TRAME

FIGURE 6
CODES D'ACCUSÉ DE RECEPTION DE TRAME

CODE ACK A TROIS BITS	INTERPRETATION
0 0 0	ZERO
0 0 1	BON
0 1 0	OCCUPE
0 1 1	REPETITION/ERREUR DETECTEE (RED)
1 0 0	ERREUR DE SEQUENCE (SEQE)
1 0 1	EXCEPTION LPI
1 1 0	EXCEPTION UNIT
1 1 1	(CODE INVALIDE/RESERVE)

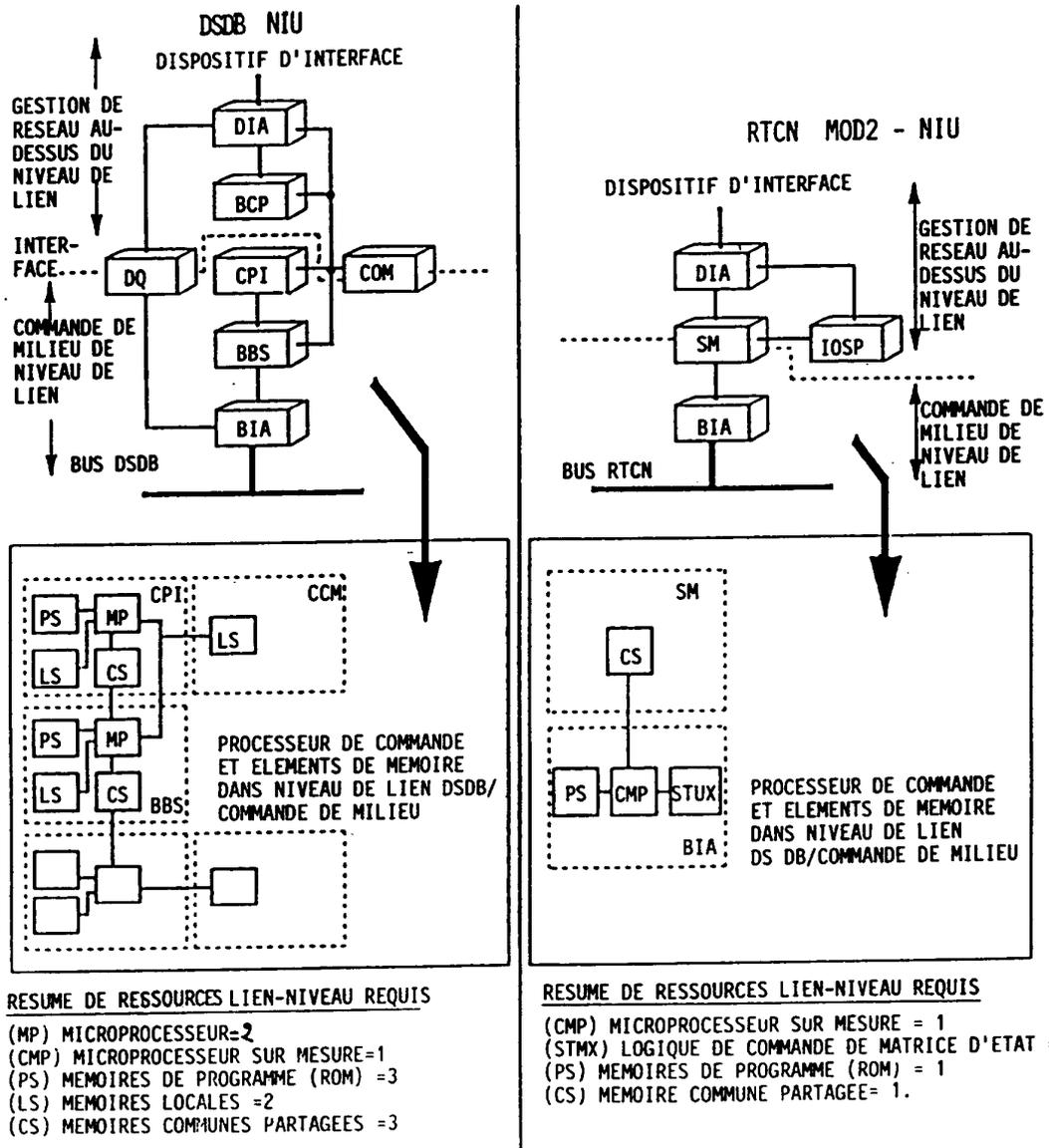
7/24

FIGURE 7.
CHAMPS DE COMMANDE DE TRAME DE BUS

MOT	PARAMETRES DE MOTS / POSITIONS DE BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	DRAPEAUX Nos								CODE DE TYPE DE TRAME (FTC)				MODIFICATEUR DE TYPE DE TRAME (FTM)			
2	VIA LPI A TRAME DIRIGEE															
3	CIBLE LPI A TRAME DIRIGEE															
4	0	0	0	0	0	0	0	0	0	0	COMPTAGE DE SOUS-PAQUETS (BAL FR)					
5	SIGNATURE DE COMMANDE DE TRAME															

FIGURE 8

COMPARAISON ENTRE L'ART ANTERIEUR ET LE SEQUEUR DE COMMANDE RTCN BIA COMPRENANT UNE LOGIQUE DE MATRICE D'ETAT



9/24

Fig. 9.

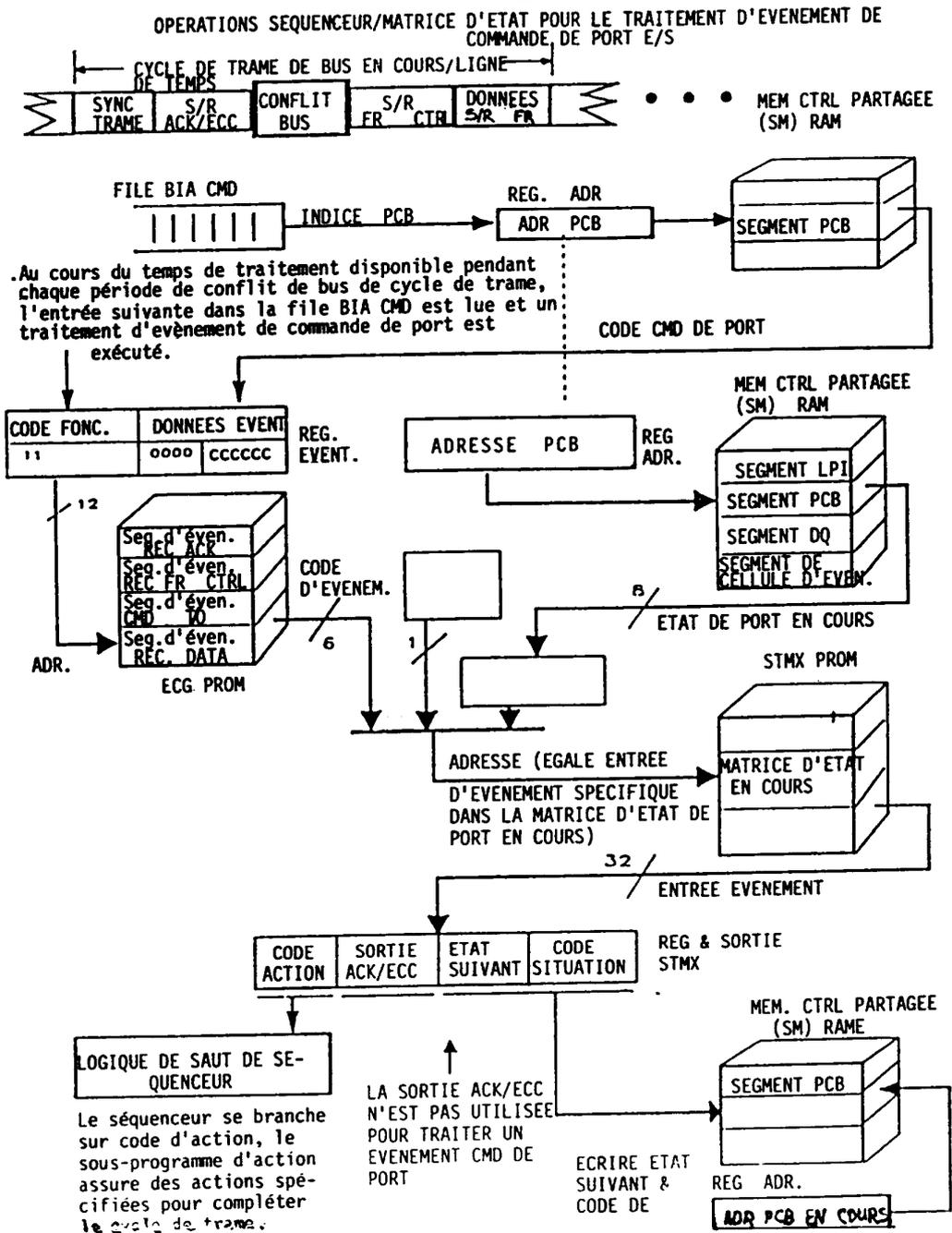
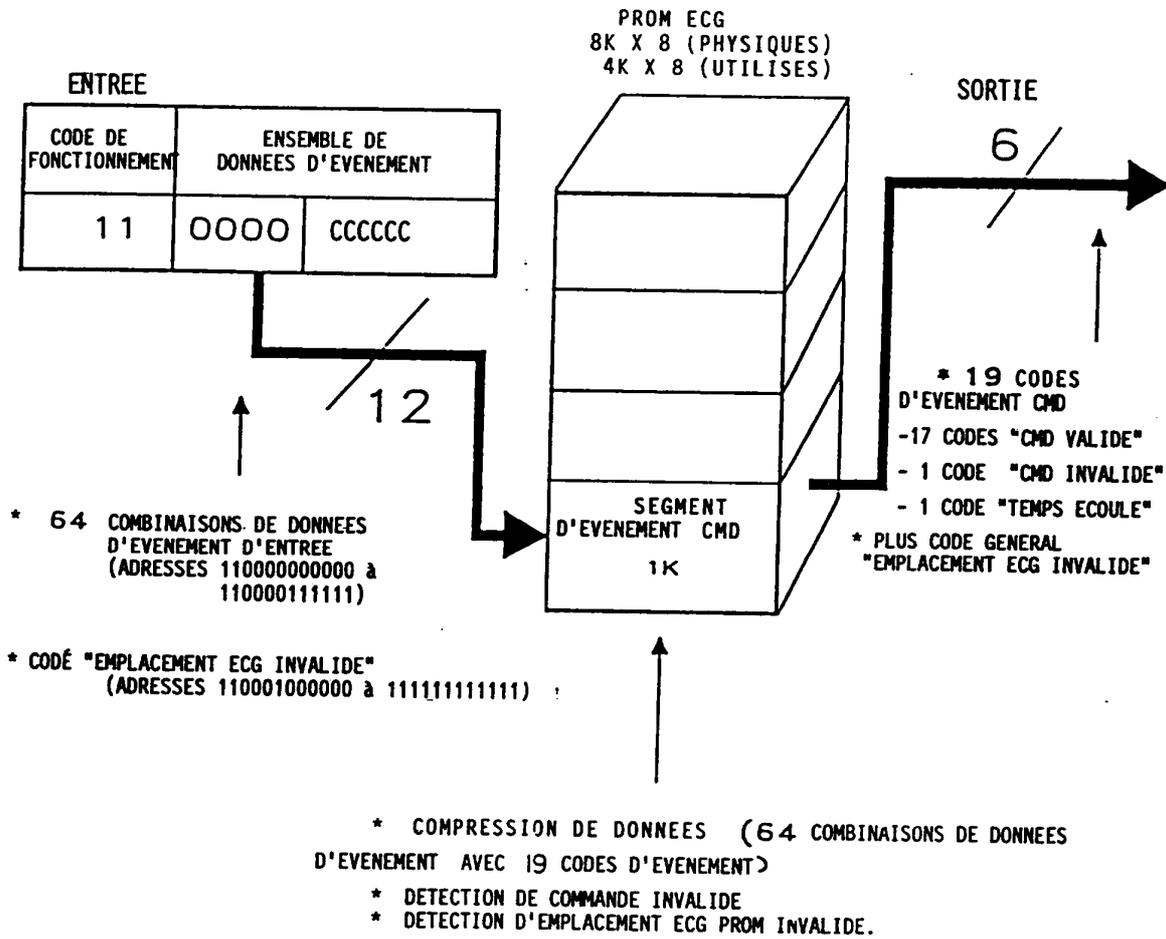


FIGURE 10

SEPARATION ET CODAGE GENEALUX DU GENERATEUR DE COTE D'EVENEMENT (ECG) PROM
POUR TRAITEMENT D'EVENEMENT DE COMMANDE DE PORT E/S



11/24

11A.

GABARIT DE CODAGE DE MATRICE D'ETAT
(page 1/3)

CODE D'EVENEMENT	MNEMONIQUE-EVENEMENT	ENTREE/SORTIES D'EVENEMENT DE MATRICE D'ETAT			
		ETAT SUIVANT	SITUATION	ACTION	ACK SORTIE
00	GOOD_ACK	<			
01	RED_ACK		EVENEMENT DE CODE D'ACCUSE DE TRAME DE RECEPTION		
02	NULL_ACK				
03	BAD_ACK	<			
04	INVALID_EVENT_A	<	CODES D'EVENEMENT NON UTILISES (RESERVE)		
05	INVALID_EVENT_B	<			
06	TOKEN	<			
07	APERIODIC_BAL				
08	PERIODIC_BAL				
09	SIG_ & CELLAVAIL				
0A	SIGNAL_ & NOCELL				
0B	BCSIG_ & -CELLAV				
0C	BC_SIG_ & -NOCELL		EVENEMENTS DE CHAMPS DE COMMANDE DE TRAME DE RECEPTION		
0D	IMMED_RESET_CMD				
0E	IMMED_HIO_CELAV				
0F	IMMED_HIO_NOCEL				
10	IMMED_FXR_CELAV				
11	IMMED_FXR_NOCEL				
12	PURGE_ & CELLAV				
13	PURGE_ & NOCELL				

11C.

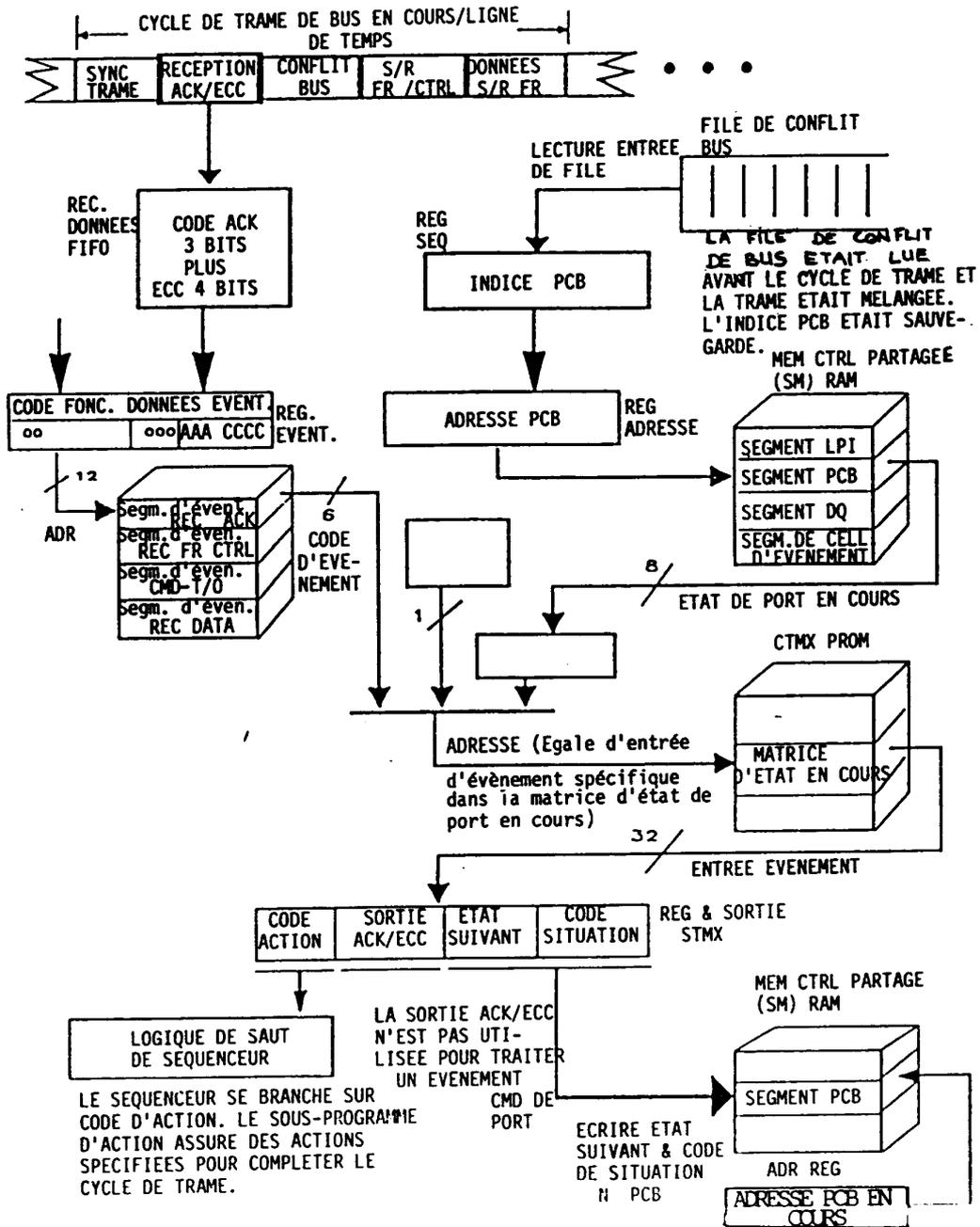
GABARIT DE CODAGE DE MATRICE D'ETAT (page 3/3)

CODE D'EVENEMENT	MNEMONIQUE-EVENEMENT	ENTREE/SORTIES D'EVENEMENT DE MATRICE D'ETAT			
		ETAT SUIVANT	SITUATION	ACTION	ACK SORTIE
2C	XMIT_BCDATA_CMD				
2D	XMIT_BCDAT/MON_CMD				
2E	EXCH_BCDAT/SIG_CMD				
2F	XMIT_BCSIG/MON_CMD				
30	EXCH_BC_SIGS_CMD				
31	XMIT_DATA_STRT_CMD				
32	XMIT_DATA/MON_CMD				
33	INVALID_EVENT_N				
34	INVALID_EVENT_O				
35	INVALID_EVENT_P				
36	INVALID_EVENT_Q				
37	INVALID_EVENT_R				
38	TIME_OUT				
39	REC_DATA_GOOD				
3A	REC_DATA_BAD				
3B	INVAL_ECG_LOCATION				
3C	INVALID_EVENT_S				
3D	INVALID_EVENT_T				
3E	INVALID_EVENT_U				
3F	INVALID_EVENT_V				

EVENEMENT DE COMMANDE DE PORT E/S (suite)			
CODE D'EVENEMENT NON UTILISE (RESERVE)			
(EVENEMENT DE TRAITEMENT CMD DE PORT E/S)			
EVENEMENT DE DONNEES DE TRAMES DE RECEPTION			
CODE D'EMPLACEMENT INVALIDE ECG PROM			
CODE D'EVENEMENT NON UTILISE (RESERVE)			

FIGURE 12.

OPERATIONS SEQUENCEUR/MATRICE D'ETAT POUR RECEVOIR DES EVENEMENTS DE CODE D'ACCUSE DE TRAME



15/24

FIGURE 13.

CODAGE ET SEPARATION GENERAUX DE LA PROM DU GENERATEUR DE CODES D'EVENEMENT (ECG) POUR TRAITEMENT D'EVENEMENT DE CODE D'ACCUSE DE TRAME DE RECEPTION.

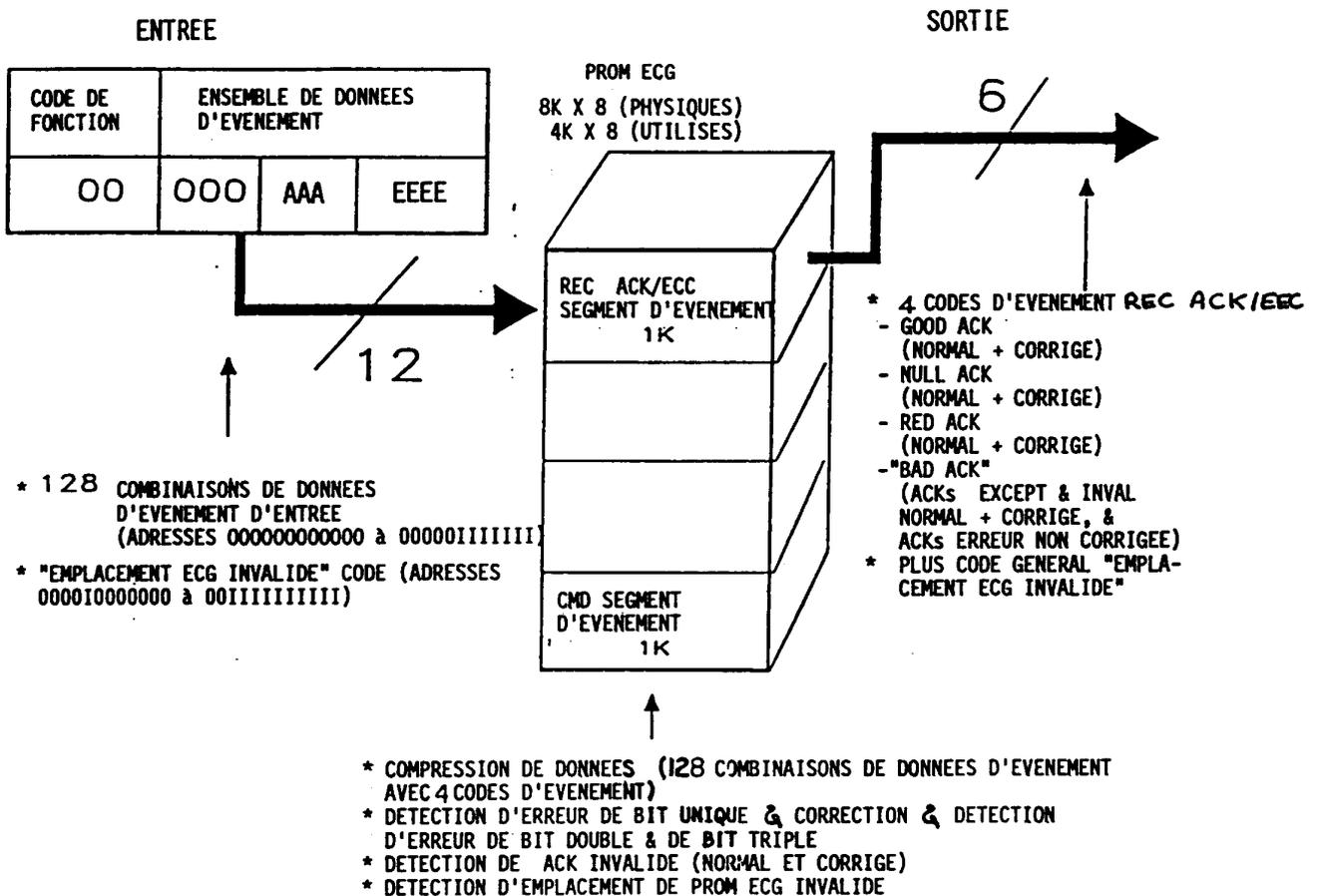


FIGURE 14

OPERATIONS SEQUEUR / MATRICE D'ETAT POUR LE TRAITEMENT D'EVENEMENT DE CHAMPS DE COMMANDE DE TRAME DE RECEPTION

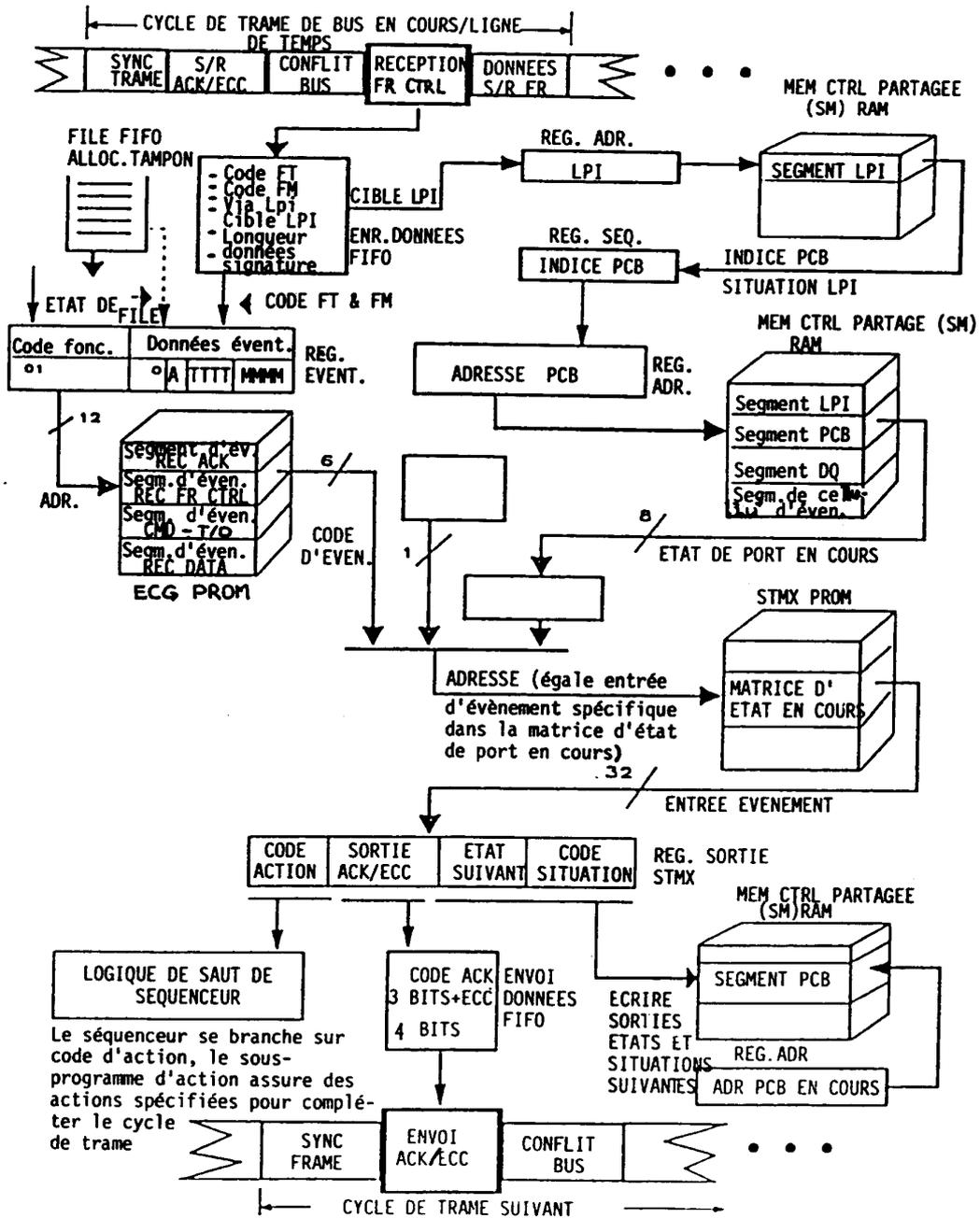


FIGURE 15.

CODAGE ET SEPARATION GENERAUX DE LA PROM DU GENERATEUR DE CODES D'EVENEMENT (ECG) POUR LE TRAITEMENT D'EVENEMENTS DE CHAMP DE COMMANDE DE TRAME DE RECEPTION.

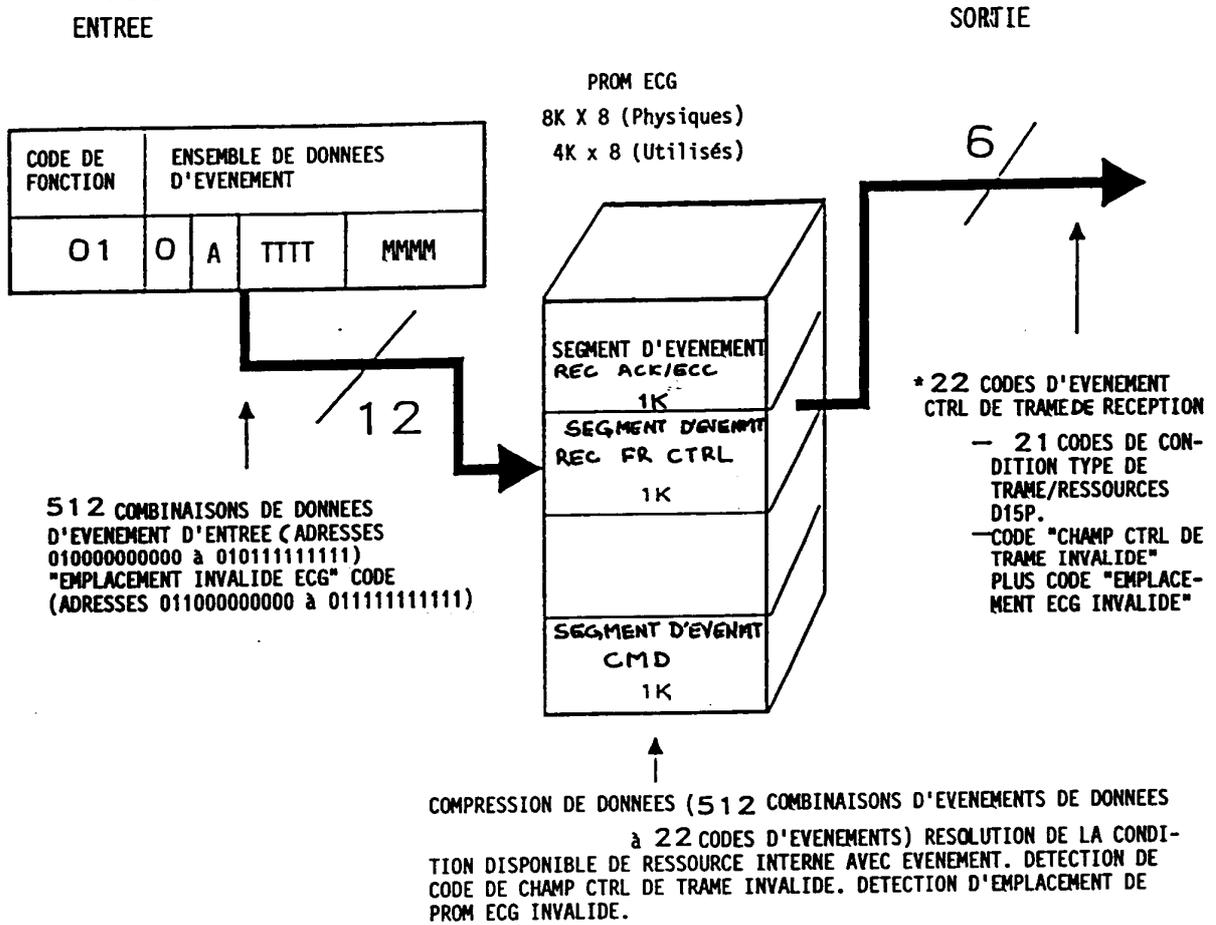


FIGURE 16.

OPERATIONS SEQUENCEUR/MATRICE D'ETAT POUR LE TRAITEMENT D'EVENEMENTS DE DONNEES DE TRAME DE RECEPTION

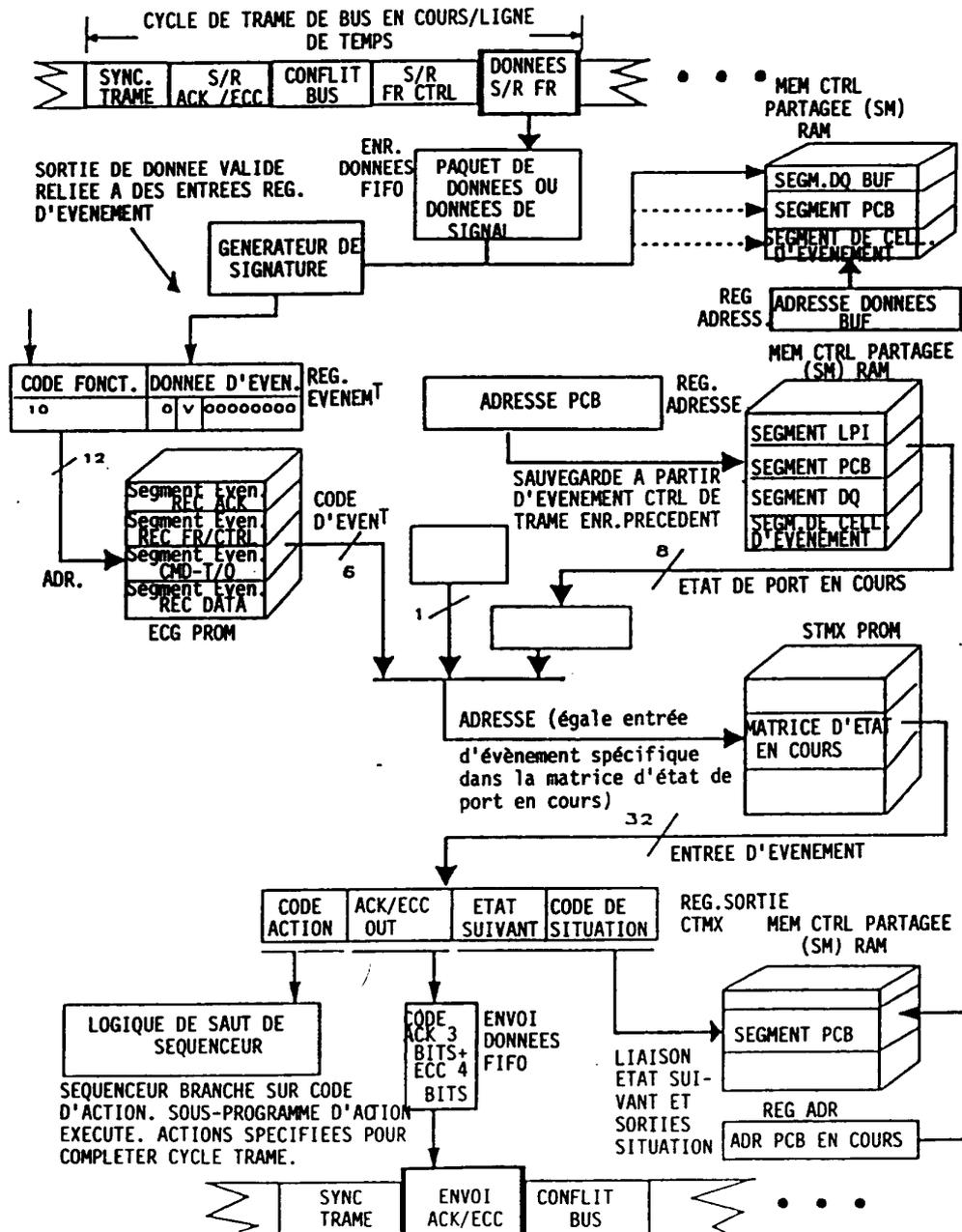


FIGURE 17.

SEPARATION ET CODAGE GENERAUX DE LA PROM DU GENERATEUR DE CODE D'EVENEMENTS (ECG) POUR LE TRAITEMENT D'EVENEMENT DE DONNEES DE TRAME DE RECEPTION.

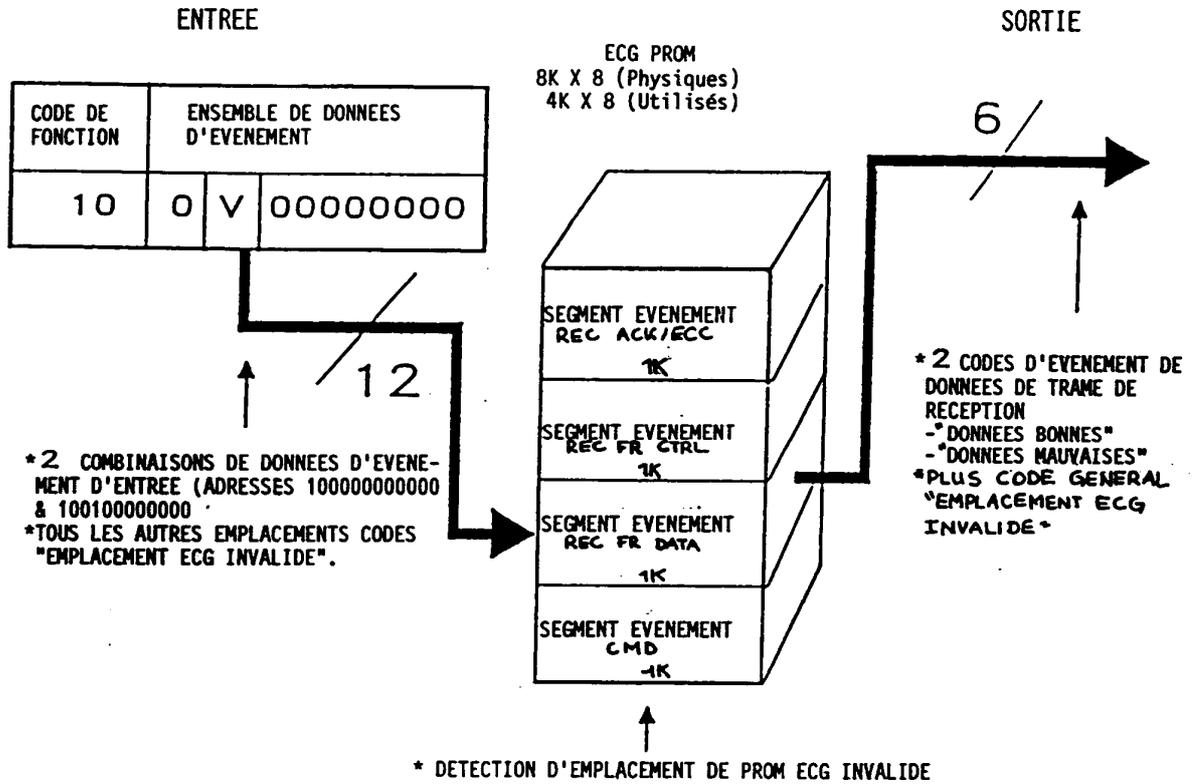
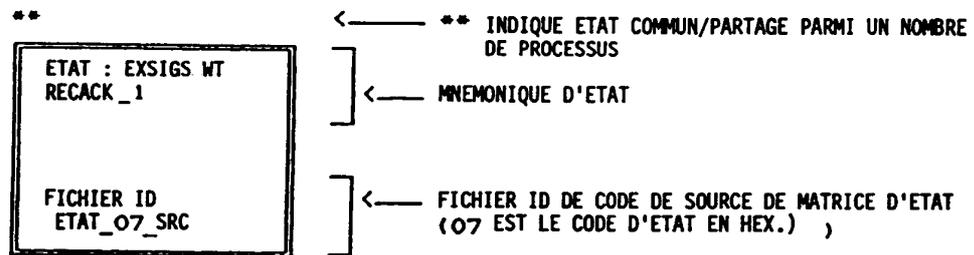
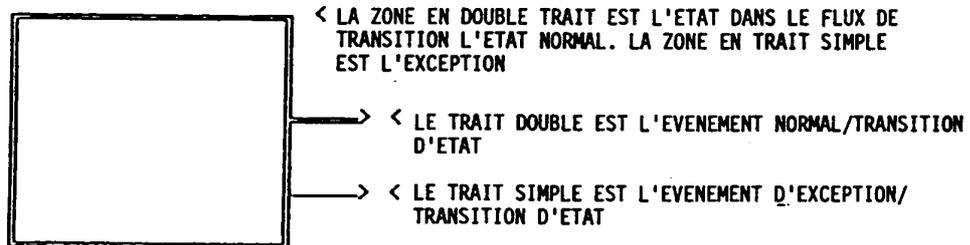


FIGURE 18.

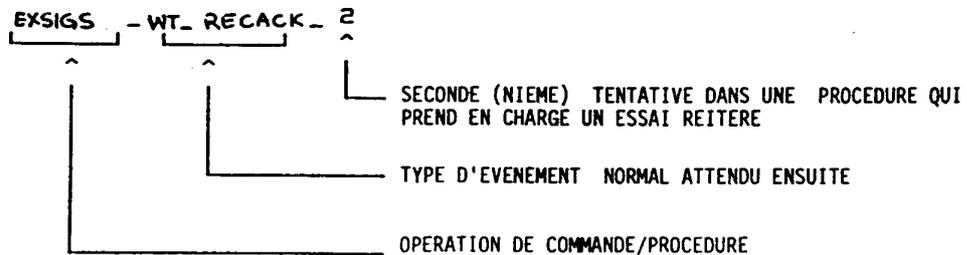
CLE POUR LES DIAGRAMMES DE TRANSITION DE NIVEAU HAUT
INFORMATION DE LIGNE INTERNE DE BLOC D'ETAT



ZONES D'ETAT ET EVENEMENT/LIGNES DE TRANSITION



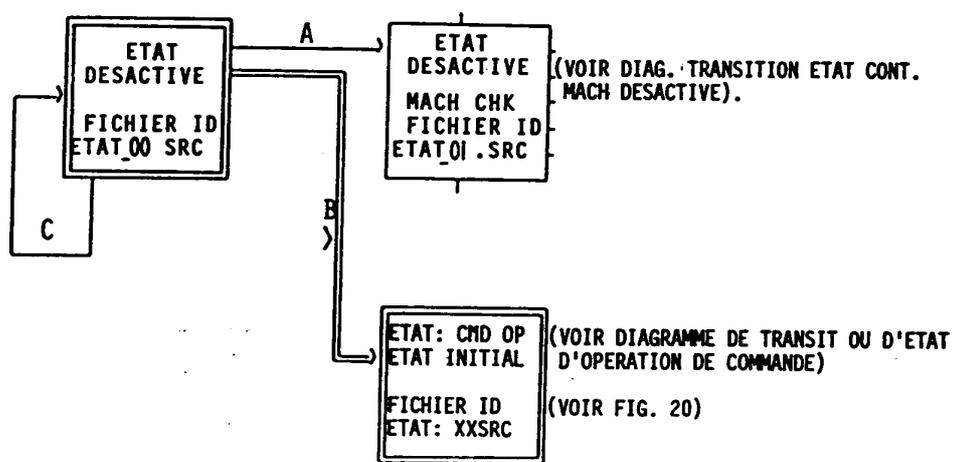
MNEMONIQUE D'ETAT (TYPE)



21/24

FIGURE 19

DIAGRAMME DE TRANSITION D'ETAT POUR : FICHER ID = SXDIAG.DIS
 ETAT DESACTIVE



(A) > CONDITION CONT. MACHINE

(B) > UNE COMMANDE BIA VALIDE

(C) > BIA DESACTIVE CMO OU CMD BIA INVALIDE OU SEQUENCE BIA CMD INVALIDE

**> ETATS COMMUN/PARTAGE, VOIR DIAGRAMMES DE TRANSITION D'ETAT ASSOCIES

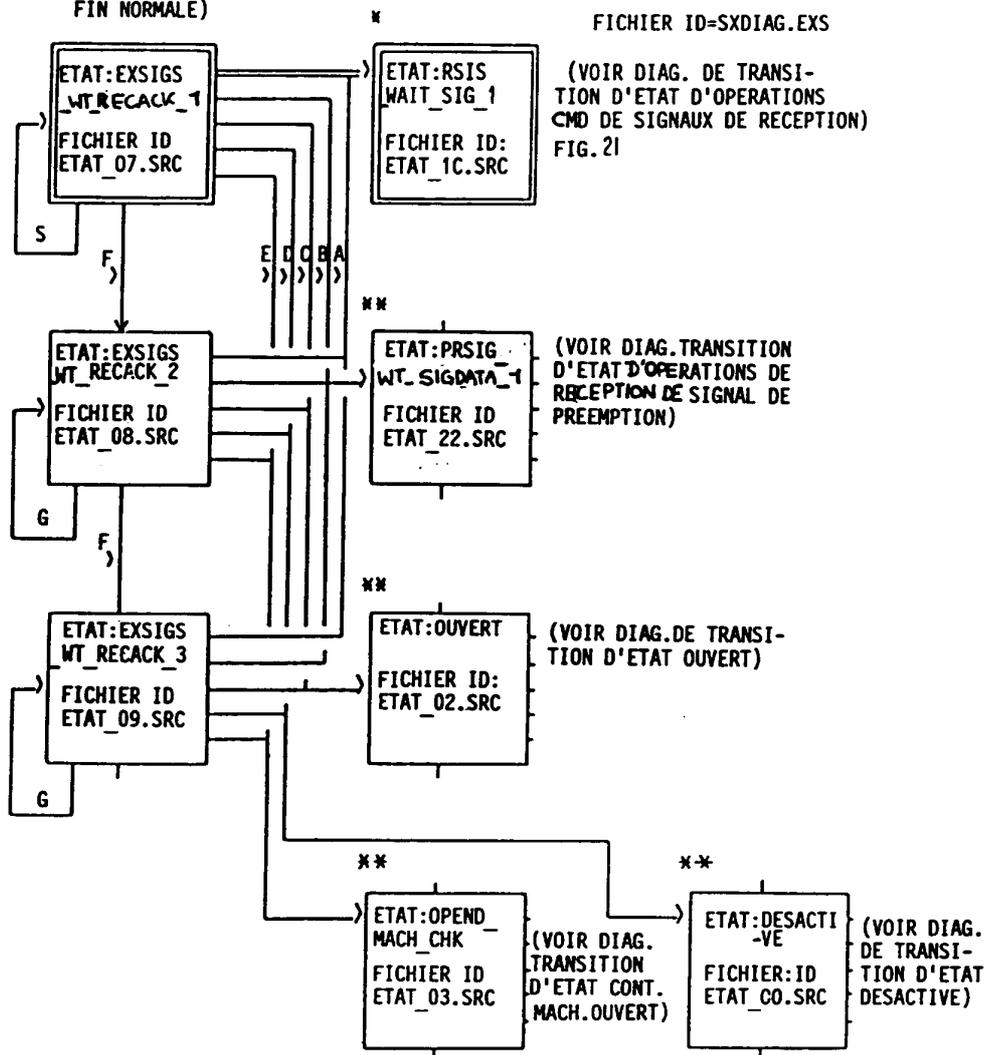
(C) > BIA DESACTIVE CMD OU BIA CMD INVALIDE OU SEQUENCE BIA CMD INVALIDE

**> ETATS COMMUN/PARTAGE, VOIR DIAGRAMMES DE TRANSITION D'ETAT ASSOCIES.

22/24

FIGURE 20.

DIAGRAMME DE TRANSITION D'ETAT POUR OPERATIONS DE COMMANDE DE SIGNAUX D'ECHANGE (CE CMD MODIFIAIT CE NIVEAU DE REVISION POUR ALLER VERS L'ETAT DE SIGNAL DE CONTROLE HORS DE FIN NORMALE)



- (A)) GOOD ACK RECU
- (B)) TRAME CTRL DE SIGNAL DE PREEMPTION RECUE
- (C)) TEMPS ECOULE, CMD BIA (CONDITION PC), BAD ACK RECU, RED ACK OU NULL ACK RECU DANS LE CAS DE EXSIGs WT_RECACK 3.
- (D)) SIGNAL CMD DE PURGE IMMEDIATE RECU
- (E)) CONDITION DE CONTROLE MACHINE
- (F)) RED ACK OU NULL ACK RECU

22.
