

19) RÉPUBLIQUE FRANÇAISE  
 INSTITUT NATIONAL  
 DE LA PROPRIÉTÉ INDUSTRIELLE  
 PARIS

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

2 866 448

21) N° d'enregistrement national :

04 05789

51) Int Cl<sup>7</sup> : G 06 F 12/08, G 06 F 13/10

12)

DEMANDE DE BREVET D'INVENTION

A1

22) Date de dépôt : 28.05.04.

30) Priorité : 16.02.04 JP 04038459.

43) Date de mise à la disposition du public de la demande : 19.08.05 Bulletin 05/33.

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) : HITACHI LTD — JP.

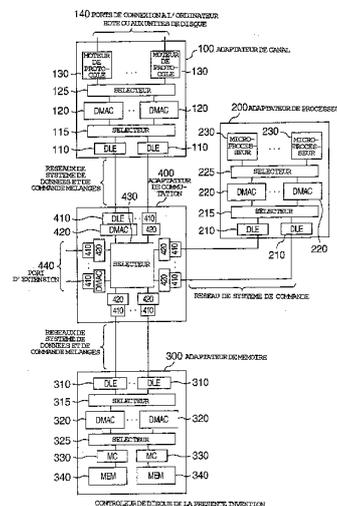
72) Inventeur(s) : HOSOYA MUTSUMI, WATANABE NAOKI, NAKAMURA SHUJI, INOUE YASUO et FUJIMOTO KAZUHISA.

73) Titulaire(s) :

74) Mandataire(s) : NONY & ASSOCIES.

54) CONTROLEUR DE DISQUE.

57) La présente invention concerne un contrôleur de disque ayant un adaptateur de canal (100) ayant une interface de connexion à un ordinateur hôte ou une unité de disque, un adaptateur de mémoire (300) mémorisant temporairement des données à transférer entre l'hôte et le disque, un adaptateur de processeur (200) commandant les opérations de l'adaptateur (100) et l'adaptateur (300), et un adaptateur de commutation (400) configurant un réseau interne en interconnectant les adaptateurs (100, 300, 200), les adaptateurs (100, 300, 200, 400) incluant chacun un contrôleur DMAC (120, 220, 320, 420) pour effectuer une commande de protocole de communication du réseau interne, et une communication multiplex de paquet est effectuée entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400). Le contrôleur de disque peut atteindre une efficacité de transfert élevée et un coût faible tout en garantissant une fiabilité élevée.



FR 2 866 448 - A1



La présente invention concerne un contrôleur de disque pour commander pour une pluralité d'unités de disque, et plus particulièrement un contrôleur de disque à fiabilité élevée utilisant une communication multiplex de type sans connexion.

Les Brevets des Etats-Unis N° 6 601 134 et N° 2 003 046 460 décrivent un système de mémorisation.

Un sous-système de disque (par la suite appelé simplement "sous-système") utilisant des unités de disque magnétique en tant que supports de mémorisation a des performances d'entrée/sortie inférieures de trois quarts par rapport à celles d'une mémoire principale d'un ordinateur utilisant des mémoires à semi-conducteurs en tant que supports de mémorisation. Beaucoup d'efforts ont été menés pour réduire cette différence, c'est-à-dire améliorer les performances d'entrée/sortie du sous-système. L'un des procédés permettant d'améliorer les performances d'entrée/sortie du sous-système consiste à utiliser un contrôleur de disque qui commande une pluralité d'unités de disque magnétique dans lesquelles des données sont mémorisées d'une manière répartie.

Par exemple, un contrôleur de disque connu de manière classique, tel que celui représenté sur la figure 16, a une pluralité d'adaptateurs de canal 2100 qui exécutent un transfert de données entre un ordinateur hôte et une unité de disque ; une pluralité d'adaptateurs de mémoire cache 2300 pour temporairement mémoriser des données à transférer entre l'ordinateur hôte et l'unité de disque ; une pluralité d'adaptateurs de mémoire de commande 2301 pour mémoriser des informations de commande concernant le fonctionnement du contrôleur de disque ; et une pluralité d'adaptateurs de commutation 2400 pour établir des connexions entre les adaptateurs de mémoire cache et des adaptateurs de canal. Les adaptateurs de canal 2100 et les adaptateurs de mémoire cache 2300 sont inter-

connectés par un réseau interne de système de données via les adaptateurs de commutation 2400. Les adaptateurs de canal 2100 et les adaptateurs de mémoire de commande 2301 sont interconnectés par un réseau interne de système de commande. A l'aide de ces connexions réseau, tous les adaptateurs de canal 2100 peuvent accéder aux adaptateurs de mémoire cache 2300 et aux adaptateurs de mémoire de commande 2301.

Chaque adaptateur de canal 2100 a : des moteurs de liaison de données (DLE) 2110 pour exécuter un transfert de paquet dans le réseau interne de système de données, des contrôleurs d'accès direct à la mémoire (DMAC) 2120 pour exécuter un transfert DMA dans le réseau interne de système de données, un sélecteur 2115 pour interconnecter les moteurs de liaison de données (DLE) 2110 et les contrôleurs DMAC 2120, des moteurs de protocole (PE) 2130 pour commander la communication entre l'ordinateur hôte et l'unité de disque, des ports 2140 pour réaliser la connexion à l'ordinateur hôte ou à l'unité de disque, des moteurs DLE 2110 pour exécuter un transfert de paquet dans le réseau interne de système de commande, des contrôleurs DMAC 2220 pour réaliser un transfert DMA dans le réseau interne de système de commande, des microprocesseurs (MP) 2230 pour commander le fonctionnement du contrôleur de disque, et un sélecteur 2125 pour interconnecter les contrôleurs DMAC 2120 et les moteurs de protocole (PE) 2130 ou les microprocesseurs 2230.

L'adaptateur de mémoire cache 2300 et l'adaptateur de mémoire de commande 2301 ont chacun : des moteurs DLE 2310 pour exécuter un transfert DMA dans le réseau interne de système de données ou le réseau interne de système de commande, des contrôleurs DMAC 2320 pour exécuter un transfert DMA dans chaque réseau interne, des contrôleurs de mémoire (MC) 2330, des modules de mémoire (MM) 2340, un sélecteur 2315 pour interconnecter les mo-

teurs DLE 2310 et les contrôleurs DMAC 2320, et un sélec-  
teur 2325 pour interconnecter les contrôleurs DMAC 2320  
et les contrôleurs MC 2330.

L'adaptateur de commutation 2400 a : des mo-  
5 teurs DLE 2410 pour exécuter un transfert de paquet dans  
le réseau interne de système de données, des contrôleurs  
DMAC 2420 pour exécuter un transfert DMA dans le réseau  
interne de système de données, et un sélecteur 2430 pour  
interconnecter les contrôleurs DMAC 2420.

10 Le transfert de données entre les adaptateurs  
est réalisé par des opérations coopératives de contrô-  
leurs DMAC dans les adaptateurs respectifs. A titre  
d'exemple, en se reportant aux figures 18 et 19, on va  
maintenant décrire une opération de déroulement d'un  
15 transfert DMA de données depuis l'ordinateur hôte vers  
l'adaptateur de mémoire cache 2300 du contrôleur de dis-  
que.

Lorsqu'une demande d'ECRITURE est envoyée de-  
puis l'ordinateur hôte via le port de connexion 2140, le  
20 microprocesseur 2230 calcule une zone de l'adaptateur de  
mémoire cache pour temporairement mémoriser les données  
D'ECRITURE, et notifie le résultat calculé au contrôleur  
DMAC 2120 dans l'adaptateur de canal sous la forme d'une  
liste DMA 2600. Le contrôleur DMAC 2120 envoie des deman-  
25 des 2605 pour acquérir des trajets jusqu'aux adaptateurs  
de mémoire cache nécessaires pour le transfert DMA. Du  
fait que les données D'ECRITURE sont mémorisées dans une  
pluralité d'adaptateurs de mémoire cache (deux adapta-  
teurs de mémoire cache ayant un contrôleur DMAC 2321 et  
30 un contrôleur DMAC 2322) afin d'améliorer la fiabilité,  
une pluralité de trajets établissant des demandes sont  
envoyés. Après que les trajets nécessaires aient été éta-  
blis, le contrôleur DMAC 2120 transfère les données  
D'ECRITURE vers le contrôleur DMAC 2420 au niveau du com-  
35 mutateur de point de relais, selon le contenu de la liste

DMA 2600. Dans ce cas, les données D'ECRITURE sont transférées depuis l'ordinateur hôte en les divisant en une quantité de données ayant une taille prédéterminée.

L'accès DMA 2420 de l'adaptateur de commutation  
5 2400 génère des sous-demandes DMA 2611 et 2612 pour les  
contrôleurs DMAC 2321 et 2322 des adaptateurs de mémoire  
cache, conformément aux demandes de transfert envoyées  
par le contrôleur DMAC 2120 de l'adaptateur de canal  
2100. En réponses aux demandes 2611 et 2612, les contrô-  
10 leurs DMAC 2321 et 2322 retournent des sous-états 2621 et  
2622 qui sont les notifications de fin de demande. Après  
que le contrôleur DMAC 2120 de l'adaptateur de canal ait  
confirmé les sous-états 2621 et 2622, il envoie la  
sous-demande DMA suivante. Lorsque les sous-états de tou-  
15 tes les sous-demandes DMA sont retournés, le contrôleur  
DMAC 2120 envoie des demandes de libération 2625 des tra-  
jets établis aux adaptateurs de mémoire cache, et re-  
tourne un état de fin 2630 au microprocesseur 2230 pour  
achever ainsi le processus relatif à la liste DMA 2600.  
20 Durant le transfert DMA, le microprocesseur 2230 accède à  
l'adaptateur de mémoire de commande 2301 lorsque néces-  
saire. Dans ce cas, un transfert DMA similaire est effec-  
tué entre le contrôleur DMAC 2220 de l'adaptateur de ca-  
nal 2100 et le contrôleur DMAC 2320 de l'adaptateur de  
25 mémoire de commande 2301.

La figure 17 représente la structure d'un pa-  
quet utilisée pour le transfert DMA. Un paquet d'instruc-  
tion 2520 a : un champ d'adresse 2521 pour indiquer un  
contrôleur DMAC cible, un champ d'adresse 2522 pour indi-  
30 quer un contrôleur DMAC de lancement, des champs  
d'adresse de mémoire 2523 et 2524 pour indiquer les  
adresses de mémoire dans lesquelles des données de trans-  
fert sont mémorisées, et un code de contrôle d'erreur  
2525.

La demande d'établissement de trajet 2605 est envoyée en utilisant un paquet d'instruction 2520. Un paquet de données 2530 a : un champ d'adresse 2531 pour indiquer un contrôleur DMAC cible, un champ d'adresse 2532 pour indiquer un contrôleur DMAC de lancement, des données de transfert 2533, et un code de contrôle d'erreur 2535. La sous-demande DMA est envoyée en utilisant le paquet de données 2530.

La figure 20 illustre un protocole de transfert de l'instruction de demande de trajet 2605 et de la sous-demande DMA 2610. Afin de faciliter un processus de rétablissement après anomalie, les processus sont tous exécutés par une communication non-multiplex. A savoir, après que l'on ait confirmé que le sous-état 2620 de la sous-demande DMA 2610 est retourné, la sous-demande suivante 2610 est envoyée.

Comme décrit ci-dessus, le transfert DMA dans un contrôleur de disque classique décrit dans les documents des brevets cités ci-dessus est réalisé par une communication non-multiplex de type à connexion du fait de sa facilité d'implémentation. C'est-à-dire que le contrôleur DMAC établit les trajets nécessaires pour l'exécution du transfert DMA, et durant le transfert DMA, les trajets sont occupés (communication de type connexion). De plus, jusqu'à ce que le sous-état relatif au sous-transfert DMA immédiatement avant ait été confirmé, la sous-demande DMA suivante ne peut être exécutée (communication non-multiplex).

Un contrôleur de disque classique a par conséquent un faible rendement d'utilisation des trajets de réseau interne, ce qui peut constituer un obstacle à l'amélioration des performances. Afin de satisfaire aux conditions selon lesquelles la largeur de bande des trajets nécessaires est réservée au rendement d'utilisation de trajet limité, une configuration de réseau interne

compliquée est requise telle qu'une implémentation à la fois du réseau interne de système de données et du réseau interne de système de commande, engendrant en résultat un coût élevé.

5                   Un but de la présente invention consiste à fournir un contrôleur de disque utilisant une communication multiplex de type sans connexion, capable de gérer les problèmes de la technique antérieure, réalisant un rendement de transfert élevé (performance) tout en retenant une fiabilité élevée équivalente à celle d'un  
10 contrôleur de disque classique, et engendrant un coût faible.

                  Afin de résoudre les problèmes décrits ci-dessus, la présente invention adopte la configuration  
15 suivante.

                  Un contrôleur de disque inclut : un adaptateur de canal ayant une interface de connexion à un ordinateur hôte ou une unité de disque, un adaptateur de mémoire pour temporairement mémoriser des données à transférer  
20 entre l'ordinateur hôte et l'unité de disque, un adaptateur de processeur pour commander les opérations de l'adaptateur de canal et de l'adaptateur de mémoire, et un adaptateur de commutation pour configurer un réseau interne en interconnectant l'adaptateur de canal, l'adaptateur de mémoire et l'adaptateur de processeur, dans lequel : l'adaptateur de canal, l'adaptateur de mémoire, l'adaptateur de processeur et l'adaptateur de commutation  
25 incluent chacun un contrôleur DMAC pour exécuter une commande de protocole de commutation du réseau interne, et une communication de multiplexage de paquet est effectuée  
30 entre les contrôleurs DMAC fournis dans les adaptateurs.

                  Selon la présente invention, en adoptant une communication multiplex de type sans connexion, le multiplexage devient possible non seulement durant un  
35 sous-transfert DMA (comme ceci sera décrit ultérieure-

ment, l'état de transfert du sous-DMA et le sous-état re-  
présentés sur la figure 11) mais également durant une  
pluralité de transferts sous-DMA (un autre état de trans-  
fert de sous-DMA 615 et sous-DMA 616 représentés sur la  
5 figure 11). L'efficacité d'utilisation des trajets peut  
être améliorée d'une manière considérable et il n'est pas  
nécessaire de séparément fournir un réseau interne de  
système de commande et un réseau interne de système de  
données comme dans le cas d'un contrôleur de disque clas-  
10 sique.

En conséquence, l'adaptateur de mémoire cache  
et l'adaptateur de mémoire de commande sont intégrés dans  
un adaptateur de mémoire. Du fait que l'efficacité d'uti-  
lisation de trajet est améliorée, la limitation d'utili-  
15 sation de trajet est relâchée de sorte que le processeur  
de l'adaptateur de canal peut être utilisé dans l'adapta-  
teur de processeur qui est indépendant de l'adaptateur de  
canal. Un contrôleur de disque peut être réalisé, lequel  
contrôleur a des performances élevées et un coût faible  
20 et a une évolutivité excellente.

La présente invention va maintenant être mieux  
comprise à partir de la lecture de la description qui va  
suivre faite en référence aux dessins annexés, sur les-  
quels :

- 25 - la figure 1 est un schéma représentant la  
structure globale d'un contrôleur de disque selon un mode  
de réalisation de la présente invention,  
- la figure 2 est un schéma représentant un  
exemple de la structure spécifique d'un moteur de liaison  
30 de données utilisée par chaque adaptateur du contrôleur  
de disque selon le mode de réalisation,  
- la figure 3 est un schéma représentant un  
exemple de la structure spécifique d'un contrôleur DMAC  
utilisée pour chaque adaptateur du contrôleur de disque  
35 selon le mode de réalisation,

- la figure 4 est un schéma représentant la structure d'un adaptateur de canal du contrôleur de disque selon le mode de réalisation,

5 - la figure 5 est un schéma représentant la structure d'un adaptateur de processeur du contrôleur de disque selon le mode de réalisation,

- la figure 6 est un schéma représentant la structure d'un adaptateur de mémoire du contrôleur de disque selon le mode de réalisation,

10 - la figure 7 est un schéma représentant la structure d'un adaptateur de commutation du contrôleur de disque selon le mode de réalisation,

15 - la figure 8 est un schéma représentant la structure d'un paquet utilisée par le contrôleur de disque selon le mode de réalisation,

- la figure 9 est un schéma illustrant un flux de paquets utilisé par le contrôleur de disque selon le mode de réalisation,

20 - la figure 10 est un schéma illustrant un protocole utilisé par le contrôleur de disque selon le mode de réalisation,

- la figure 11 est un schéma illustrant un protocole de transfert de communication multiplex utilisé par le contrôleur de disque selon le mode de réalisation,

25 - la figure 12 est un schéma illustrant un déroulement de mise à jour de champ de séquence DMA durant une sous-transmission DMA utilisée par le contrôleur de disque selon le mode de réalisation,

30 - la figure 13 est un schéma illustrant un déroulement de confirmation de champ de séquence DMA durant une réception de sous-état utilisée par le contrôleur de disque selon le mode de réalisation,

35 - la figure 14 est un schéma représentant la structure globale d'un contrôleur de disque selon un autre mode de réalisation de la présente invention,

- la figure 15 est un schéma représentant la structure globale d'un contrôleur de disque selon encore un autre mode de réalisation de la présente invention,

5 - la figure 16 est un schéma représentant la structure globale d'un contrôleur de disque classique,

- la figure 17 est un schéma représentant la structure d'un paquet utilisée par le contrôleur de disque classique,

10 - la figure 18 est un schéma illustrant un flux de paquets utilisé par le contrôleur de disque classique,

- la figure 19 est un schéma illustrant un protocole utilisé par le contrôleur de disque classique, et

15 - la figure 20 est un schéma illustrant un protocole de communication non-multiplex utilisé par le contrôleur de disque classique.

On va maintenant décrire en détail des modes de réalisation d'un contrôleur de disque de la présente invention en se reportant aux figures 1 à 15.

20 La figure 1 est un schéma représentant la structure globale d'un contrôleur de disque selon un mode de réalisation de la présente invention. Le contrôleur de disque du présent mode de réalisation a : un adaptateur de canal 100 ayant une interface 140 destinée à être connectée à un ordinateur hôte ou à une unité de disque, un  
25 adaptateur de mémoire 300 pour temporairement mémoriser des données à transférer entre l'ordinateur hôte et l'unité de disque, un adaptateur de processeur 200 pour commander les opérations de l'adaptateur de canal 100 et d'un adaptateur de mémoire 300, et un adaptateur de com-  
30 mutation 400 constituant un réseau interne en interconnectant l'adaptateur de canal 100, l'adaptateur de mémoire 300 et l'adaptateur de processeur 200.

L'adaptateur de canal 100, l'adaptateur de processeur 200, l'adaptateur de mémoire 300 et adaptateur de  
35 commutation 400 ont des contrôleurs DMAC (DMAC) 120, 220,

320 et 420, respectivement, les contrôleurs DMAC exécutant une commande de protocole de communication du réseau interne. Les adaptateurs de commutation peuvent être connectés les uns aux autres par leurs ports d'extension  
5 440. Les contrôleurs DMAC exécutent un transfert DMA avec l'implication de moteurs de liaison de données (DLE) 110, 210, 310 et 410, respectivement. La communication multiplex de paquets de type sans connexion représentée sur la figure 11 est effectuée entre ces contrôleurs DMAC.

10 La figure 11 est un schéma illustrant un protocole de transfert de communication multiplex utilisé par le contrôleur de disque selon le mode de réalisation de la présente invention. Comme représenté sur la figure 11, sans la confirmation d'un sous-état relatif à une  
15 sous-demande DMA, la sous-demande DMA suivante est envoyée (communication multiplex, c'est-à-dire une communication multiplex durant un sous-transfert DMA). De plus, le transfert DMA compris entre DMA1 et DMA2 et un transfert compris entre DMA3 et DMA4 partagent le même trajet  
20 entre les moteurs DLE1 et DLE2 (communication de type sans connexion). Dans l'exemple représenté sur la figure 11, un sous-DMA 615 et un sous-DMA 616 sont d'une manière alternée transférés en partageant le même trajet entre les moteurs DLE1 et DLE2 pour effectuer une communication  
25 multiplex. Comme ceci sera compris d'après la description de la figure 8 qui sera citée ultérieurement, la communication multiplex de type sans connexion devient possible en adoptant la structure de paquet qui contient des informations (IDENTIFICATEUR DE TACHE) pour la commande de  
30 séquence d'une destination, des données et une sous-demande DMA.

Dans l'exemple représenté sur les figures 1 et 11, l'adoption de la communication multiplex de type sans connexion permet un multiplexage non seulement durant un  
35 sous-transfert DMA mais également durant des

sous-transferts DMA. Par conséquent, l'efficacité d'utilisation de trajet peut être fortement améliorée (du fait que le transfert de données peut être effectué sans intervalle de temps entre les trajets). Il est quasi inutile d'implémenter de manière séparée le réseau interne de système de commande et le réseau interne de système de données, comme ceci est réalisé d'une manière classique. Il est par conséquent possible d'utiliser l'adaptateur de mémoire intégrant l'adaptateur de mémoire cache et l'adaptateur de mémoire de commande, et d'utiliser de plus l'adaptateur de processeur indépendamment de l'adaptateur de canal du fait que la limite d'utilisation de trajet est relâchée. Un contrôleur de disque ayant un coût faible et une excellente évolutivité et souplesse peut par conséquent être réalisé.

La figure 5 est un schéma représentant un exemple de la structure spécifique d'un adaptateur de processeur du contrôleur de disque selon le mode de réalisation de la présente invention, et la figure 2 est un schéma représentant la structure spécifique d'un moteur de liaison de données utilisée par l'adaptateur de processeur. La structure du moteur de liaison de données (DLE) représentée sur la figure 2 peut être appliquée non seulement à l'adaptateur de processeur mais également à d'autres adaptateurs.

L'adaptateur de processeur 200 représenté sur la figure 5 a : des microprocesseurs (MP) 230, une pluralité de contrôleurs DMAC 220 et un ou plusieurs moteurs de liaison de données (DLE) 210. Un sélecteur 225 interconnecte les microprocesseurs 230 et les contrôleurs DMAC 220, et une pluralité de contrôleurs DMAC 220 partagent les moteurs DLE 210 via le sélecteur 215. A savoir, le nombre de contrôleurs DMAC est habituellement beaucoup plus grand que le nombre de moteurs DLE.

Du fait qu'un arbitre DMA 2150 du sélecteur 215 arbitre les demandes provenant d'une pluralité de contrôleurs DMAC 220, un transfert DMA à partir d'une pluralité de contrôleurs DMAC via le même moteur DLE 210 peut être  
5 exécuté en même temps (communication sans connexion). Des données de réception provenant du moteur DLE 210 sont distribuées par un arbitre DLE 2155 à un contrôleur DMAC cible 220.

Comme représenté sur la figure 2, le moteur DLE  
10 a un port de transmission 1101, un tampon de transmission 1102, un port de réception 1105, un tampon de réception 1106, une logique de relance 1110 et un tampon de relance 1120. Le tampon de relance et la logique de relance exécutent un processus pour réaliser un transfert sans erreur  
15 au niveau liaison de données. C'est-à-dire qu'un paquet envoyé depuis le tampon de transmission au port de transmission est mémorisé dans le tampon de relance 1120 par la logique de relance 1110. Un état représentatif indiquant si le paquet est arrivé correctement est retourné  
20 au port de réception, et si une erreur est notifiée, le paquet est à nouveau envoyé depuis le tampon de relance par la logique de relance. La structure de moteur DLE représentée sur la figure 2 permet un contrôle d'erreur de liaison de données dans l'unité de paquet et réalise une  
25 communication multiplex.

Dans l'exemple de la structure représentée sur les figures 5 et 2, une communication multiplex de type sans connexion devient possible et un contrôleur de disque peut être réalisé, lequel contrôleur a des performances élevées et est souple et simple et a un coût faible.  
30

La figure 4 est un schéma représentant un exemple de la structure spécifique de l'adaptateur de canal du contrôleur de disque selon le mode de réalisation de la présente invention, et la figure 3 est un schéma représentant un exemple de la structure spécifique du con-  
35

contrôleur DMAC utilisée par l'adaptateur de canal. La structure du contrôleur DMAC (DMAC) représentée sur la figure 3 est applicable non seulement à l'adaptateur de canal mais également à d'autres adaptateurs.

5 L'adaptateur de canal représenté sur la figure 4 a des moteurs de protocole 130, des contrôleurs DMAC 120 et des moteurs DLE 110. Le moteur PE 130 et le contrôleur DMAC 120 sont connectés par un sélecteur 125, et le contrôleur DMAC 120 et le moteur DLE 110 sont connectés par un sélecteur 115. Chaque contrôleur DMAC 120 a une pluralité de tampons de réception de type premier-entré premier-sorti (FIFO) VC0 et VC1 et une pluralité de tampons FIFO de transmission VC0 et VC1.

10 Le contrôleur DMAC 120 représenté sur la figure 3 est constitué d'un multiplexeur 1201, de tampons FIFO de transmission 1202, d'un démultiplexeur 1205, de tampons FIFO de réception 1206, d'une logique de transaction 1210, d'une table de gestion de séquence 1220, d'une logique d'assemblage de paquet 1230 et d'une logique de désassemblage de paquet 1240. Un arbitre 1212 arbitre le conflit de données de transmission entre une pluralité de tampons FIFO de transmission 1202 et le multiplexeur 1201 sélectionne les données de transmission.

15 D'une manière similaire, le démultiplexeur 1205 sélectionne des données de réception sous la commande de l'arbitre 1212 et les mémorise dans un tampon FIFO correct parmi une pluralité de tampons FIFO de réception 1206. La logique d'assemblage de paquet 1230 et la logique de désassemblage de paquet 1240 sont des circuits logiques servant à assembler et à désassembler le paquet. La logique de commande de séquence 1213 et la table de gestion de séquence 1220 gèrent la séquence DMA des sous-transferts DMA, la description de cette opération étant ultérieurement mentionnée.

20

25

30

Dans l'exemple représenté sur les figures 4 et 3, une pluralité de tampons VC0 et VC1 peuvent être utilisés pour chaque moteur DLE. Par exemple, un moteur DLE peut utiliser un mélange du réseau interne de système de commande et du réseau interne de système de données (par exemple, VC0 est utilisé pour le réseau interne de système de données, et VC1 est utilisé pour le réseau de système de commande). L'arbitre 1212 peut agir pour affecter un ordre de priorité à une pluralité de tampons. Par exemple, si le réseau interne de système de commande est défini de manière à avoir la priorité sur le réseau interne de système de données, il est possible d'éviter une plus longue durée de délai d'accès du réseau interne de système de commande autrement provoquée par un mélange des deux réseaux. A savoir, à l'aide de cet agencement, il est possible d'obtenir un contrôleur de disque ayant une configuration réseau interne plus simple et à la fois une amélioration des performances et un coût bas.

La figure 6 est un schéma représentant un exemple de la structure spécifique de l'adaptateur de mémoire du contrôleur de disque selon le mode de réalisation de la présente invention. L'adaptateur de mémoire représenté sur la figure 6 a des modules de mémoire (MM) 340, des contrôleurs de mémoire (MC) 330, des contrôleurs DMAC 320 et des moteurs DLE 310. Les contrôleurs de mémoire 330 et les contrôleurs DMAC 320 sont interconnectés par un sélecteur 325, et le contrôleur DMAC 320 et le moteur DLE 310 sont interconnectés par un sélecteur 315. Chaque contrôleur DMAC (DMAC) 320 a un tampon de réception (VC0 ou VC1) et un tampon de transmission (VC0 ou VC1). Le conflit entre données de transmission est arbitré entre une pluralité de tampons FIFO de transmission VC0 et entre une pluralité de tampons FIFO de transmission VC1 pour transférer des données vers le moteur DLE 310. D'une manière similaire, le conflit entre données de réception

est arbitré entre une pluralité de tampons FIFO de réception VC0 et entre une pluralité de tampons FIFO de réception VC1 pour mémoriser des données dans une file FIFO de réception correcte.

5 Les arbitres 3250 et 3255 arbitrent les conditions de conflit entre le contrôleur DMAC 320 et le contrôleur de mémoire 330. Un contrôleur de mémoire peut par conséquent être partagé par une pluralité de contrôleurs DMAC, et la commande d'ordre de priorité entre le  
10 contrôleur DMAC peut être réalisée en fonction des arbitres. Par exemple, si des contrôleurs DMAC du réseau interne de système de commande et des contrôleurs DMAC du réseau interne de système de données sont fournis et les  
15 contrôleurs DMAC du réseau interne de système de commande sont définis de manière à avoir la priorité sur le réseau interne de système de données, alors les accès au réseau interne de système de commande peuvent être moins influencés par les interférences du fonctionnement du réseau interne de système de données.

20 Avec la structure représentée sur la figure 6, une pluralité de contrôleurs DMAC peuvent être utilisés en correspondance avec un moteur DLE. Par exemple, un moteur DLE a un mélange du réseau interne de système de commande et du réseau interne de système de données. Une  
25 pluralité de contrôleurs DMAC peuvent être utilisés en correspondance avec un contrôleur de mémoire en permettant un mélange constitué de la mémoire de système de commande et de la mémoire de système de données. Avec cette structure, par conséquent, il devient possible de  
30 réaliser un contrôleur de disque ayant une structure de réseau interne plus simple, satisfaisant à la fois à l'amélioration des performances et à la réduction des coûts.

35 La figure 8 est un schéma représentant un exemple de la structure spécifique du paquet à transférer en-

tre une pluralité de contrôleurs DMAC dans le contrôleur de disque selon le mode de réalisation de la présente invention. Le paquet 500 représenté sur la figure 8 a au moins un champ d'adresse 511 pour indiquer un contrôleur DMAC cible, un champ d'adresse 521 pour indiquer un contrôleur DMAC de lancement et un champ de séquence DMA 524 pour gérer la séquence de transfert lorsqu'un transfert DMA est divisé en une pluralité de paquets.

Dans le contrôleur de disque selon le mode de réalisation de la présente invention, du fait que le transfert DMA est effectué par une communication multiplex de type sans connexion, il est nécessaire de garantir la séquence de transfert d'accès DMA et d'effectuer correctement un processus de contrôle d'erreur et un processus de récupération après panne. Comme moyens permettant cela, le champ de séquence DMA est fourni de manière à identifier d'une manière fiable le paquet, et ce champ est contrôlé (de préférence séquentiellement incrémenté) de manière à ce qu'il soit unique (pouvant être distingué) dans un seul transfert DMA.

A l'aide de l'exemple de la structure de paquet représentée sur la figure 8, une garantie de séquence correcte et son contrôle sont possibles dans le transfert DMA par une connexion multiplex du type sans connexion, et un processus de reprise après défaillance peut être effectué lorsqu'une défaillance se produit. Avec cette structure, il devient possible de réaliser un contrôleur de disque ayant une fiabilité élevée équivalente à la fiabilité d'un contrôleur de disque classique.

Le paquet 500 représenté sur la figure 8 a une première adresse 511 pour désigner un contrôleur DMAC de relais de paquet, des deuxième et troisième adresses 522 et 523 pour désigner des contrôleurs DMAC cibles et des données de transfert 531 destinées à être transférées vers les contrôleurs DMAC cibles. Lorsqu'une demande

D'ECRITURE est envoyée depuis l'adaptateur de canal 100 à l'adaptateur de mémoire 300, la première adresse désigne le contrôleur DMAC 420 de l'adaptateur de commutation et les deuxième et troisième adresses désignent des contrôleurs DMAC 320 de l'adaptateur de mémoire. Une pluralité d'adresses des adaptateurs de mémoire sont désignées afin d'améliorer la fiabilité effectuant une ECRITURE dupli-  
5 quée pour les mémoires cache.

A l'aide de cette structure de paquet, la fonction de transfert DMA incluant L'ECRITURE dupliquée peut être appliquée à une communication multiplex sans connexion, ce qui fait que le contrôleur de disque de fiabilité élevée peut être réalisé.  
10

Le paquet 500 représenté sur la figure 8 a également un en-tête de routage 510 contenant des informations de commande pour le moteur DLE, un en-tête d'instruction 520 contenant des informations de commande pour le contrôleur DMAC, et un bloc de données 530 contenant d'autres données. L'en-tête de routage 510 a un code de contrôle d'erreur d'en-tête de routage 515 pour contrôler toute erreur de transfert dans l'en-tête de routage. L'en-tête d'instruction 520 a un code de contrôle d'erreur d'en-tête d'instruction 525 pour contrôler toute erreur de transfert dans l'en-tête d'instruction. Le bloc de données 530 a un code de contrôle d'erreur de bloc de données 535 pour contrôler toute erreur de transfert dans le bloc de données.  
15  
20  
25

A l'aide de cette structure de paquet, les informations de commande de routage, les informations de commande DMAC et les informations de données peuvent être protégées par différents codes de contrôle d'erreur, engendrant en résultat une commande de transfert DMA plus précise et un processus de rétablissement après défaillance plus précis. Même si les informations de commande de routage doivent être réécrites comme lorsqu'une ECRI-  
30  
35

TURE dupliquée est effectuée via l'adaptateur de commutation, il est possible de minimiser la plage de recalcul du code de contrôle d'erreur et de réaliser le contrôleur de disque ayant une fiabilité élevée et des performances élevées.

5

La figure 9 est un schéma représentant le flux d'un paquet utilisé par le contrôleur de disque selon le mode de réalisation de la présente invention, et la figure 10 est un schéma illustrant un protocole utilisé par le contrôleur de disque selon le mode de réalisation de la présente invention. Dans l'exemple représenté sur les figures 9 et 10, une sous-demande DMA 610 est envoyée depuis le contrôleur DMAC 120 de l'adaptateur de canal au contrôleur DMAC 420 de l'adaptateur de commutation. Dans le paquet de la sous-demande DMA 610, le champ d'adresse de lancement 521 désigne le contrôleur DMAC d'adaptateur de canal 120 en tant que DMA maître et le champ d'adresse cible 511 désigne le contrôleur DMAC d'adaptateur de commutation 420.

10

15

20

Le contrôleur DMAC 420 renvoie des sous-états de fin 621 et 622 correspondant à la sous-demande de transfert DMA 610 au contrôleur DMAC 120. Les sous-états de fin 621 et 622 contiennent les informations du champ de séquence DMA 524 contenu dans la sous-demande DMA 610. Le contrôleur DMAC 120 confirme les informations de ce champ de séquence DMA afin de confirmer la séquence de transfert des sous-transferts DMA.

25

30

La figure 12 est un schéma illustrant un ordi-nogramme de mise à jour de champ de séquence DMA durant la sous-transmission DMA utilisée par le contrôleur de disque selon le mode de réalisation de la présente invention, et la figure 13 est un schéma illustrant un ordi-nogramme de confirmation de champ de séquence DMA durant la réception de sous-état utilisée par le contrôleur de disque selon le mode de réalisation de la présente inven-

35

tion. Chaque contrôleur DMAC maintient la valeur d'un champ de séquence DMA courant dans une variable CURR\_DMA\_SEQ. Durant la sous-transmission DMA, pendant que CURR\_DMA\_SEQ est incrémentée, elle est insérée dans le champ de séquence DMA 524 de chaque paquet de transfert. Chaque contrôleur DMAC maintient la valeur du sous-état DMA à retourner ensuite, dans une variable NEXT\_DMA\_SEQ. Lorsque le sous-état DMA est retourné, la valeur de la séquence DMA est comparée à une valeur prévue. Si les deux coïncident l'une avec l'autre, la valeur coïncidant NEXT\_DMA\_SEQ est incrémentée. Si les deux ne coïncident pas, les sous-demandes de transfert DMA en cours d'exécution (à partir de NEXT\_DMA\_SEQ vers CURR\_DMA\_SEQ) sont annulées et par la suite une défaillance est notifiée au processeur.

Dans l'exemple de la structure représentée sur les figures 9 et 10 et les figures 12 et 13, également pour le sous-transfert DMA, la séquence de transfert de chaque accès DMA peut être commandée de manière fiable en utilisant le champ de séquence DMA 524. C'est-à-dire qu'avec cette structure, un contrôleur de disque ayant une fiabilité élevée peut être réalisé en utilisant une communication multiplex sans connexion.

Les figures 9 et 10 illustrent également un flux de paquet (protocole) D'ECRITURE double utilisé par le contrôleur de disque selon le mode de réalisation de la présente invention. Dans cet exemple de la structure, les sous-demandes DMA 611 et 612 sont envoyées par le contrôleur DMAC d'adaptateur de canal 120 aux contrôleurs DMAC d'adaptateur de mémoire 321 et 322 via le contrôleur DMAC d'adaptateur de commutation 420. Dans le paquet de la sous-demande DMA 610, le champ d'adresse de lancement 521 désigne le contrôleur DMAC d'adaptateur de canal 120, le champ d'adresse cible 511 désigne le contrôleur DMAC d'adaptateur de commutation 420, le champ cible 511 dési-

gne les contrôleurs DMAC d'adaptateur de mémoire 321 et 322, et le bloc de données (champ) 531 mémorise les données de transfert.

Le contrôleur DMAC 420 de l'adaptateur de commutation génère un paquet de sous-demande DMA 611 et un  
5 paquet de sous-demande DMA 612 et transfère les paquets aux adresses cibles respectives. Le premier paquet 611 a le contrôleur DMAC 321 en tant que champ d'adresse cible et contient les données de transfert 531, et le deuxième  
10 paquet 611 a le contrôleur DMAC 321 en tant que champ d'adresse cible et contient les données de transfert 531. En réponses aux sous-demandes DMA 611 et 612, les contrôleurs DMAC 321 et 322 de l'adaptateur de mémoire retournent les sous-états 621 et 622 au contrôleur DMAC d'adaptateur de canal 120 via le contrôleur DMAC d'adaptateur  
15 de commutation 420.

L'exemple de la structure représentée sur les figures 9 et 10 peut réaliser L'ECRITURE double en mémoire cache par le contrôleur DMAC d'adaptateur de commutation. Du fait que le contrôleur DMAC 420 de l'adaptateur de commutation 400 près de l'adaptateur de mémoire  
20 300 génère les paquets pour une ECRITURE double, la largeur de bande du réseau interne ne sera pas consommée avec gaspillage et l'efficacité du trajet peut être améliorée. A l'aide de l'exemple de cette structure, un contrôleur de disque ayant des performances élevées et une  
25 fiabilité élevée peut être réalisé.

La figure 7 est un schéma représentant un exemple de la structure spécifique de l'adaptateur de commutation du contrôleur de disque selon le mode de réalisation de la présente invention. L'adaptateur de commutation représenté sur la figure 7 a une pluralité de moteurs DLE 410, une pluralité de contrôleurs DMAC 420 et un sélecteur 430. Un paquet reçu depuis le moteur DLE côté  
30 réception 410 est mémorisé d'une manière répartie dans  
35

une pluralité de tampons FIFO de réception (VC0, VC1) dans un contrôleur DMAC côté réception 420, et par la suite, envoyé à des tampons FIFO de transmission d'un contrôleur DMAC de transmission 420 via des circuits logiques de sélection 4301, 4302, 4306 et 4307 préparés pour les tampons FIFO de transmission respectifs, et transmis depuis un moteur DLE côté transmission 410.

A l'aide de l'exemple de la structure représentée sur la figure 7, d'une manière similaire au paquet ayant les informations de commande de routage, les informations de commande DMAC et les informations de données représentées sur la figure 8, un paquet à transférer entre une pluralité de contrôleurs DMAC a un en-tête incluant des informations DMAC cible et un champ de données incluant d'autres données. L'en-tête inclut un code de contrôle d'erreur d'en-tête pour contrôler toute erreur de transfert dans l'en-tête. Le champ de données inclut un code de contrôle d'erreur de champ de données pour contrôler toute erreur de transfert dans le champ de données.

Jusqu'à ce que le code de contrôle d'erreur d'en-tête soit confirmé, le contrôleur DMAC côté réception 420 de l'adaptateur de commutation ne va pas envoyer le paquet au contrôleur DMAC côté transmission. Après que le code de contrôle d'erreur d'en-tête ait été confirmé, l'en-tête et le champ de données du paquet sont envoyés au contrôleur DMAC côté transmission suivant un traitement en pipeline. Si une erreur est trouvée par le code de contrôle d'erreur d'en-tête, le paquet est rejeté et un processus de correction d'erreur correct est exécuté.

A l'aide de l'exemple de la structure représentée sur la figure 7, l'adaptateur de commutation peut lancer un processus de transmission depuis le moteur DLE de transmission avant que le champ de données entier soit sondé depuis le moteur DLE de réception et le code de

contrôle d'erreur de champ de données soit confirmé, et le paquet ayant un champ d'adresse cible illégal du fait d'une erreur dans l'en-tête est rejeté pour éviter la propagation de l'erreur. A l'aide de l'exemple de la structure, un contrôleur de disque ayant des performances élevées et une fiabilité élevée peut être réalisé.

L'adaptateur utilisé par le contrôleur de disque selon le mode de réalisation de la présente invention, tel que l'adaptateur de canal représenté sur la figure 4 et l'adaptateur de processeur représenté sur la figure 5, a la structure où une pluralité de contrôleurs DMAC partagent une pluralité de moteurs DLE. Dans le cas de l'adaptateur de canal représenté sur la figure 4, deux moteurs DLE et seize contrôleurs DMAC sont prévus et il peut exister le cas où chaque contrôleur DMAC partage quelques moteurs DLE. Avec cette structure de redondance, par exemple, durant une commutation DMA par le contrôleur DMAC via un certain moteur DLE, même si une défaillance survient dans ce moteur DLE, l'arbitre DMAC 1150 (faire référence à la figure 4) ou 2150 (faire référence à la figure 5) effectue une commande de routage pour connecter un autre moteur DLE. D'une manière similaire, l'arbitre DMAC 1150 ou 2150 effectue une commande de routage pour une pluralité de processus DMAC pour distribuer les processus à une pluralité de moteurs DLE et réaliser une distribution de charge.

A l'aide de l'exemple de la structure, l'arbitre DMAC 1150 ou 2150 effectue une commande pour faire en sorte que le même moteur DLE traite une transmission/réception pour une série de sous-demandes DMA et sous-états provenant du même contrôleur DMAC. D'une manière plus préférée, une transmission/réception dans le mode de fonctionnement normal est fixée pour les demandes et les sous-états du même contrôleur DMAC.

A l'aide de l'exemple de la structure représentée sur les figures 4 et 5, l'acheminement de réseau interne est fixé pour une série de sous-demandes DMA et sous-états. Par conséquent, il n'y a pas de possibilité d'échange de séquence (dégagement) dû à des acheminements différents. La commande de séquence des sous-demandes DMA et des sous-états peut être fortement facilitée. Notamment, avec l'exemple de la structure, un contrôleur de disque ayant une fiabilité élevée peut être réalisé facilement.

La figure 14 est un schéma représentant la structure globale d'un contrôleur de disque selon un autre mode de réalisation de la présente invention. Dans ce mode de réalisation de la présente invention représenté sur la figure 14, une pluralité d'adaptateurs de canal 100, une pluralité d'adaptateurs de processeur 200 et une pluralité d'adaptateurs de mémoire 300 sont interconnectés par une pluralité d'adaptateurs de commutation 400. En fournissant une pluralité de trajets entre tous les adaptateurs, il devient possible de réaliser une redondance capable d'une reprise après défaillance en un point arbitraire. La connexion de chaque adaptateur est comme représentée sur la figure 14. Chaque adaptateur a les trajets pour deux adaptateurs correspondants.

Selon le présent mode de réalisation de la présente invention, la fiabilité peut être améliorée en améliorant la redondance du système du contrôleur de disque.

La figure 15 est un schéma représentant la structure globale d'un contrôleur de disque selon encore un autre mode de réalisation de la présente invention. Dans le présent mode de réalisation de la présente invention représenté sur la figure 16, deux contrôleurs de disque du mode de réalisation représenté sur la figure 14 sont utilisés en couplant des ports d'extension des adaptateurs de commutation. A l'aide de cette connexion, des

adaptateurs de canal, des adaptateurs de processeur et des adaptateurs de mémoire supplémentaires peuvent être installés de sorte que l'évolutivité du système peut être améliorée en utilisant la même architecture. A l'aide de ce mode de réalisation de la présente invention, l'évolutivité du contrôleur de disque peut être améliorée.

Comme décrit jusqu'à présent, l'adoption du contrôleur de disque des modes de réalisation de la présente invention représentés sur les figures 1 à 15 peut fournir les fonctions et résultats suivants. Selon les modes de réalisation, une pluralité de tampons peuvent être établis dans une correspondance bi-univoque avec un moteur DLE. Par exemple, le réseau interne de système de commande et le réseau interne de système de données peuvent être mélangés dans un seul moteur DLE. L'arbitre peut définir l'ordre de priorité d'une pluralité de tampons. Par exemple, si le réseau interne de système de commande est défini de manière à avoir la priorité sur le réseau interne de système de données, il est possible d'éviter un plus long délai d'accès du réseau interne de système de commande autrement provoqué par un mélange des deux réseaux. A l'aide de cet agencement, il est possible de réaliser un contrôleur de disque ayant une configuration de réseau interne plus simple et d'obtenir à la fois l'amélioration des performances et la réduction des coûts.

Selon les modes de réalisation, une pluralité de contrôleurs DMAC peuvent être définis dans une correspondance bi-univoque avec un moteur DLE. Par exemple, le réseau interne de système de commande et le réseau interne de système de données peuvent être mélangés dans un seul moteur DLE. Une pluralité de contrôleurs DMAC peuvent être définis dans une correspondance bi-univoque avec un contrôleur de mémoire, de sorte que le réseau interne de système de commande et le réseau interne de sys-

tème de données peuvent être mélangés. Un contrôleur de disque ayant une structure de réseau interne plus simple peut être réalisé, satisfaisant à la fois à l'amélioration des performances et à la réduction des coûts.

5                    Selon les modes de réalisation, une garantie de séquence correcte et son contrôle sont possibles dans le transfert DMA par une communication multiplex de type sans connexion, et un processus de récupération après défaillance correcte peut être effectué lorsqu'une défaillance survient. A l'aide de cette structure, il devient possible de réaliser un contrôleur de disque ayant une fiabilité élevée équivalente à la fiabilité d'un contrôleur de disque classique. Selon les modes de réalisation, les informations de commande de routage, les informations de commande DMAC et les informations de données peuvent être protégées par des codes de contrôle d'erreur différents, engendrant en résultat une commande de transfert DMA plus précise et un processus de reprise après défaillance plus précis. Même si les informations de commande de routage doivent être réécrites comme lorsqu'une ECRITURE double est effectuée via l'adaptateur de commutation, il est possible de minimiser la plage de recalcul du code de contrôle d'erreur et de réaliser le contrôleur de disque ayant une fiabilité élevée et des performances élevées.

25                    Selon les modes de réalisation, il devient possible de réaliser une ECRITURE double en mémoire cache par le contrôleur DMAC d'adaptateur de commutation. Du fait que le contrôleur DMAC de l'adaptateur de commutation près de l'adaptateur de mémoire génère les paquets pour une ECRITURE double, la largeur de bande du réseau interne ne sera pas consommée avec gaspillage et l'efficacité des trajets peut être améliorée.

30                    Selon les modes de réalisation, l'adaptateur de commutation peut lancer un processus de transmission de-

35

puis le moteur DLE de transmission avant que le champ de données entier soit lu depuis le moteur DLE de réception et le code de contrôle d'erreur de champ de données soit confirmé, et le paquet ayant un champ d'adresse cible il-  
5 légal du fait d'une erreur dans l'en-tête est rejeté pour éviter la propagation de l'erreur. Selon les modes de réalisation, du fait que l'acheminement de réseau interne est fixé pour une série de sous-demandes DMA et sous-états, il n'y a pas de possibilité d'échange de sé-  
10 quence (dégagement) du fait des acheminements différents. La commande de séquence des sous-demandes DMA et des sous-états peut être fortement facilitée.

15 Selon les modes de réalisation, la fiabilité peut être améliorée en intégrant la redondance avec le système du contrôleur de disque. Selon les modes de réalisation, l'évolutivité du contrôleur de disque peut être améliorée.

REVENDICATIONS

1. Contrôleur de disque, caractérisé en ce qu'il comporte :

5 un adaptateur de canal (100) ayant une interface de connexion à un ordinateur hôte ou une unité de disque,

un adaptateur de mémoire (300) pour temporairement mémoriser des données à transférer entre l'ordinateur hôte et l'unité de disque,

10 un adaptateur de processeur (200) pour commander les opérations de l'adaptateur de canal (100) et de l'adaptateur de mémoire (300), et

un adaptateur de commutation (400) pour configurer un réseau interne en interconnectant l'adaptateur de canal (100), l'adaptateur de mémoire (300) et l'adaptateur de processeur (200),

15 dans lequel :

l'adaptateur de canal (100), l'adaptateur de mémoire (300), l'adaptateur de processeur (200) et l'adaptateur de commutation (400) incluent chacun un contrôleur d'accès direct à la mémoire (DMA) (120, 220, 320, 420) pour exécuter une commande de protocole de communication du réseau interne, et

25 une communication multiplex de paquet est effectuée entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400).

2. Contrôleur de disque selon la revendication 1, caractérisé en ce que :

30 l'adaptateur de canal (100), l'adaptateur de mémoire (300), l'adaptateur de processeur (200) et l'adaptateur de commutation (400) incluent chacun une pluralité de contrôleurs DMAC (120, 220, 320, 420) et un ou plusieurs moteurs de liaison de données (110, 210, 310, 410), et

la pluralité de contrôleurs DMAC (120, 220, 320, 420) partagent le moteur de liaison de données (110, 210, 310, 410) et effectuent un transfert DMA en même temps via le moteur de liaison de données (110, 210, 310, 410).

3. Contrôleur de disque selon la revendication 1 ou 2, caractérisé en ce que :

le contrôleur DMAC (120, 220, 320, 420) inclut une pluralité de tampons FIFO de réception (VC0, VC1) et une pluralité de tampons FIFO de transmission (VC0, VC1), et

un conflit de données de réception est arbitré entre la pluralité de tampons FIFO de réception (VC0, VC1), et un conflit de données de transmission est arbitré entre la pluralité de tampons FIFO de transmission (VC0, VC1).

4. Contrôleur de disque selon la revendication 1 ou 2, caractérisé en ce que :

le contrôleur DMAC (120, 220, 320, 420) inclut un tampon FIFO de réception (VC0, VC1) et un tampon FIFO de transmission (VC0, VC1), et

un conflit de données de réception est arbitré entre les tampons FIFO de réception respectifs (VC0, VC1) appartenant à une pluralité de contrôleurs DMAC (120, 220, 320, 420), et un conflit de données de transmission est arbitré entre des tampons FIFO de transmission respectifs (VC0, VC1) appartenant à une pluralité de contrôleurs DMAC (120, 220, 320, 420).

5. Contrôleur de disque, caractérisé en ce qu'il comporte :

un adaptateur de canal (100) ayant une interface de connexion à un ordinateur hôte ou une unité de disque,

un adaptateur de mémoire (300) pour mémoriser temporairement des données à transférer entre l'ordinateur hôte et l'unité de disque,

5 un adaptateur de processeur (200) pour commander les opérations de l'adaptateur de canal (100) et de l'adaptateur de mémoire (300), et

10 un adaptateur de commutation (400) pour configurer un réseau interne en interconnectant l'adaptateur de canal (100), l'adaptateur de mémoire (300) et l'adaptateur de processeur (200),

dans lequel :

15 l'adaptateur de canal (100), l'adaptateur de mémoire (300), l'adaptateur de processeur (200) et l'adaptateur de commutation (400) incluent chacun une pluralité de contrôleurs DMAC (120, 220, 320, 420) pour effectuer une commande de protocole de communication du réseau interne, et un ou plusieurs moteurs de liaison de données (110, 210, 310, 410) partagés par les contrôleurs DMAC (120, 220, 320, 420),

20 le contrôleur DMAC (120, 220, 320, 420) inclut une pluralité de tampons FIFO de réception (VC0, VC1) et une pluralité de tampons FIFO de transmission (VC0, VC1), un moteur de liaison de données (110, 210, 310, 410) étant mis en correspondance avec une pluralité de tampons  
25 (VC0, VC1),

30 un conflit de données de réception est arbitré entre les tampons FIFO de réception respectifs (VC0, VC1) appartenant à une pluralité de contrôleurs DMAC (120, 220, 320, 420), et un conflit de données de transmission est arbitré entre les tampons FIFO de transmission respectifs (VC0, VC1) appartenant à une pluralité de contrôleurs DMAC (120, 220, 320, 420), pour définir ainsi un ordre de priorité d'une pluralité de tampons (VC0, VC1),  
et

un réseau interne de système de commande et un réseau interne de système de données sont mélangés dans un seul moteur de liaison de données (110, 210, 310, 410), et une communication multiplex de paquet est effectuée entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400).

6. Contrôleur de disque, caractérisé en ce qu'il comporte :

un adaptateur de canal (100) ayant une interface de connexion à un ordinateur hôte ou une unité de disque,

un adaptateur de mémoire (300) pour temporairement mémoriser des données à transférer entre l'ordinateur hôte et l'unité de disque,

un adaptateur de processeur (200) pour commander les opérations de l'adaptateur de canal (100) et de l'adaptateur de mémoire (300), et

un adaptateur de commutation (400) pour configurer un réseau interne en interconnectant l'adaptateur de canal (100), l'adaptateur de mémoire (300) et l'adaptateur de processeur (200),

dans lequel :

l'adaptateur de canal (100), l'adaptateur de mémoire (300), l'adaptateur de processeur (200) et l'adaptateur de commutation (400) incluent chacun un contrôleur DMA (120, 220, 320, 420) pour exécuter une commande de protocole de communication du réseau interne,

un paquet (500) à transférer entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400) a un champ d'adresse pour désigner un contrôleur DMAC cible, un champ d'adresse (521) pour désigner un contrôleur DMAC de lancement et un champ de séquence DMA (524) pour gérer un ordre de transfert lorsqu'un transfert DMA est divisé en une pluralité de paquets (500), et

le champ de séquence DMA (524) a un identificateur de tâche unique à un transfert DMA.

7. Contrôleur de disque selon la revendication 6, caractérisé en ce qu'un paquet (500) à transférer entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400) a une première adresse (511) pour désigner un contrôleur DMAC (120, 220, 320, 420) relais du paquet, des deuxième et troisième adresses (522, 523) pour désigner des contrôleurs DMAC cibles (120, 220, 320, 420), et des données de transfert (531) à transférer aux contrôleurs DMAC cibles (120, 220, 320, 420).

8. Contrôleur de disque selon la revendication 6 ou 7, caractérisé en ce que :

l'adaptateur de canal (100), l'adaptateur de mémoire (300), l'adaptateur de processeur (200) et l'adaptateur de commutation (400) incluent chacun une pluralité de contrôleurs DMAC (120, 220, 320, 420) et un ou plusieurs moteurs de liaison de données (110, 210, 310, 410),

un paquet (500) à transférer entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400) comporte un champ de routage (510) contenant des informations de commande pour le moteur de liaison de données (110, 210, 310, 410), un champ d'instruction (520) contenant des informations de commande pour le contrôleur DMAC et un champ de données (530) contenant d'autres données, et

le champ de routage (510) inclut un code de contrôle d'erreur de champ de routage (515) pour contrôler une erreur de transfert dans le champ de routage (510), le champ d'instruction (520) inclut un code de contrôle d'erreur de champ d'instruction (525) pour contrôler une erreur de transfert dans le champ d'instruction (520), et le champ de données (530) inclut un

code de contrôle d'erreur de champ de données (535) pour contrôler une erreur de transfert dans le champ de données (530).

5 9. Contrôleur de disque selon la revendication 6, 7 ou 8, caractérisé en ce que :

un sous-transfert DMA est effectué depuis un contrôleur DMAC (120, 220, 320, 420) désigné par le champ d'adresse de lancement vers un contrôleur DMAC (120, 220, 320, 420) désigné par le champ d'adresse cible,

10 le contrôleur DMAC (120, 220, 320, 420) désigné par le champ d'adresse cible renvoie un sous-état de fin correspondant au sous-transfert DMA au contrôleur DMAC (120, 220, 320, 420) désigné par le champ d'adresse de lancement,

15 le sous-état de fin inclut des informations du champ de séquence DMA (524) contenu dans le sous-transfert DMA, et

20 le contrôleur DMAC (120, 220, 320, 420) désigné par le champ d'adresse de lancement confirme les informations du champ de séquence DMA (524) contenu dans le sous-état de fin pour confirmer ainsi une séquence de transfert du sous-transfert DMA.

10. Contrôleur de disque selon l'une quelconque des revendications 6 à 9, caractérisé en ce que :

25 si un paquet (500) à transférer entre les contrôleurs DMAC (120, 220, 320, 420) a une première adresse (511) pour désigner le contrôleur DMAC (420) dans l'adaptateur de commutation (400), des deuxième et troisième adresses (522, 523) pour désigner des contrôleurs DMAC cibles (120, 220, 320, 420), et des données de transfert à transférer vers les contrôleurs DMAC cibles (120, 220, 320, 420),

30 le contrôleur DMAC (420) dans l'adaptateur de commutation (400) génère un paquet (500) qui a la deuxième adresse (522) dans le champ d'adresse cible et

35

contient les données de transfert et un paquet (500) qui a la troisième adresse (523) dans le champ d'adresse cible et contient les données de transfert.

5 11. Contrôleur de disque selon la revendication 6 ou 7, caractérisé en ce que :

un paquet (500) à transférer entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400) comporte un champ d'en-tête (510) contenant des informations de commande de paquet et  
10 un champ de données (530) contenant d'autres données,

le champ d'en-tête (510) inclut un code de contrôle d'erreur de champ d'en-tête (515) pour contrôler une erreur de transfert dans le champ d'en-tête(510), et le champ de données (530) inclut un code de contrôle  
15 d'erreur de champ de données (535) pour contrôler une erreur de transfert dans le champ de données (530), et

le contrôleur DMAC (420) dans l'adaptateur de commutation (400) envoie uniquement un paquet (500) ayant un code de contrôle d'erreur de champ d'en-tête (515)  
20 correct.

12. Contrôleur de disque selon la revendication 1, caractérisé en ce que :

l'adaptateur de canal (100), l'adaptateur de mémoire (300), l'adaptateur de processeur (200) et  
25 l'adaptateur de commutation (400) incluent chacun une pluralité de contrôleurs DMAC (120, 220, 320, 420) et une pluralité de moteurs de liaison de données (110, 210, 310, 410), et

lorsque le contrôleur DMAC effectue un transfert DMA via le moteur de liaison de données, le transfert DMA est effectué via un même moteur de liaison de données durant un transfert DMA.  
30

13. Contrôleur de disque ayant un premier contrôleur de disque et un autre contrôleur de disque,  
35 caractérisé en ce que :

le premier contrôleur de disque comporte :

un premier adaptateur de canal (100) ayant une interface de connexion à un ordinateur hôte ou à une unité de disque,

5 un premier adaptateur de mémoire (300) pour temporairement mémoriser des données à transférer entre l'ordinateur hôte et l'unité de disque,

un premier adaptateur de processeur (200) pour commander les opérations du premier adaptateur de canal (100) et du premier adaptateur de mémoire (300), et

10 un premier adaptateur de commutation (400) pour configurer un réseau interne en interconnectant le premier adaptateur de canal (100), le premier adaptateur de mémoire (300) et le premier adaptateur de processeur (200),

15 dans lequel :

le premier adaptateur de canal (100), le premier adaptateur de mémoire (300), le premier adaptateur de processeur (200) et le premier adaptateur de commutation (400) incluent chacun un contrôleur DMAC pour effectuer une commande de protocole de communication du réseau interne, et

20 une communication multiplex de paquet est effectuée entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400), et

25 dans lequel l'autre contrôleur de disque comporte des adaptateurs ayant des structures similaires aux structures du premier adaptateur de canal (100), du premier adaptateur de mémoire (300), du premier adaptateur de processeur (200) et du premier adaptateur de commutation (400) du premier contrôleur de disque,

30 le premier adaptateur de commutation (400) est connecté à chacun des premiers adaptateurs (100, 200, 300) et à chacun des autres adaptateurs, et

l'autre adaptateur de commutation (400) est connecté à chacun des autres adaptateurs et à chacun des premiers adaptateurs (100, 200, 400).

5 14. Contrôleur de disque selon la revendication 13, caractérisé en ce que des ports d'extension (440) du premier adaptateur de commutation (400) et/ou de l'autre adaptateur de commutation sont connectés à des ports d'extension d'un autre adaptateur de commutation.

10 15. Contrôleur de disque, caractérisé en ce qu'il comporte :

un adaptateur de canal (100) ayant une interface de connexion à un ordinateur hôte ou une unité de disque,

15 un adaptateur de mémoire (300) pour temporairement mémoriser des données à transférer entre l'ordinateur hôte et l'unité de disque,

un adaptateur de processeur (200) pour commander les opérations de l'adaptateur de canal (100) et de l'adaptateur de mémoire (300), et

20 un adaptateur de commutation (400) pour configurer un réseau interne en interconnectant l'adaptateur de canal (100), l'adaptateur de mémoire (300) et l'adaptateur de processeur (200),

dans lequel :

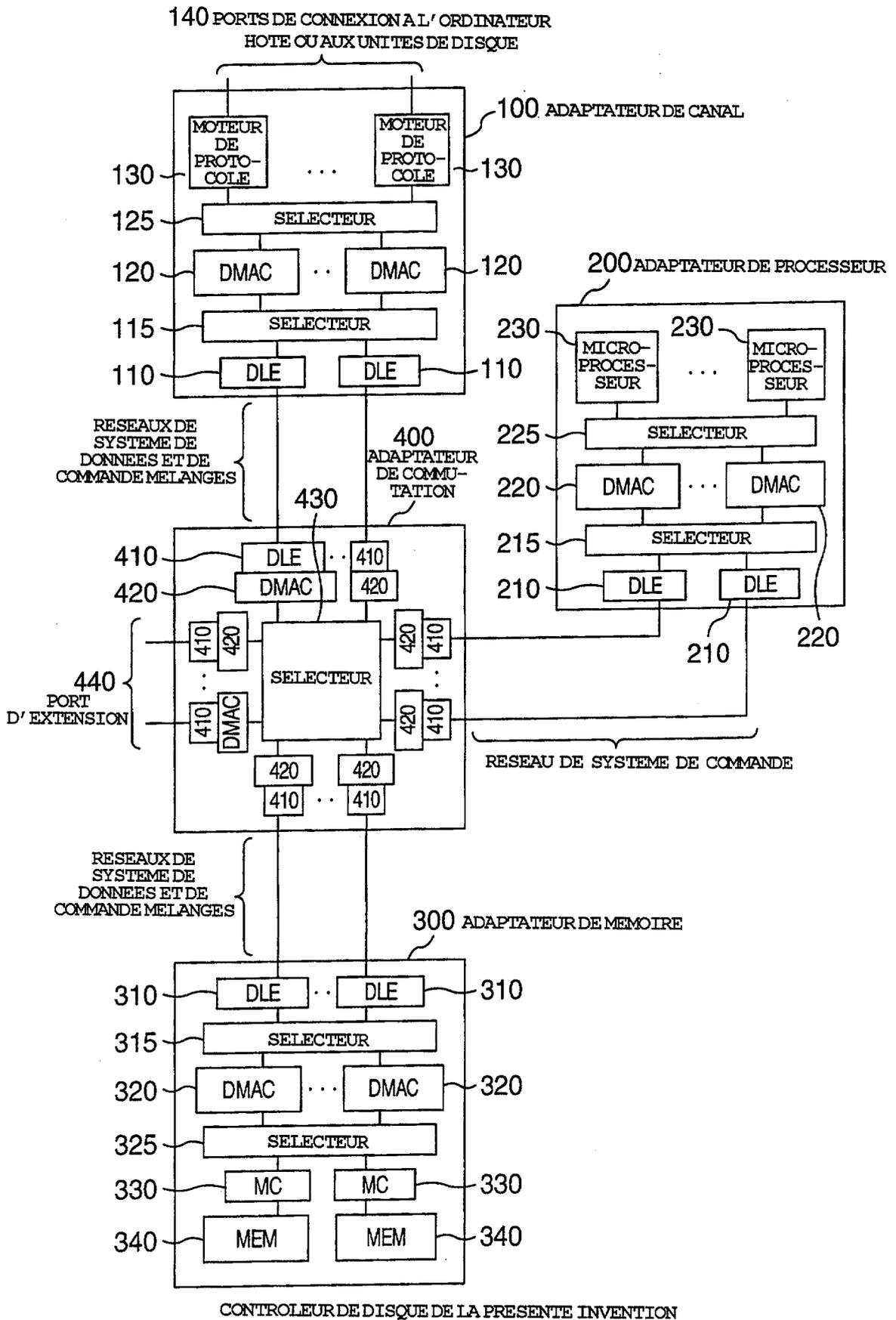
25 l'adaptateur de canal (100), l'adaptateur de mémoire (300), l'adaptateur de processeur (200) et l'adaptateur de commutation (400) incluent chacun un contrôleur DMAC pour effectuer une commande de protocole de communication du réseau interne, et un moteur de liaison de données pour exécuter un transfert DMA vers et depuis le réseau interne, et

30 un paquet (500) à transférer entre les contrôleurs DMAC (120, 220, 320, 420) fournis dans les adaptateurs (100, 200, 300, 400) comporte un champ d'adresse pour désigner un contrôleur DMAC cible (120, 220, 320,

35

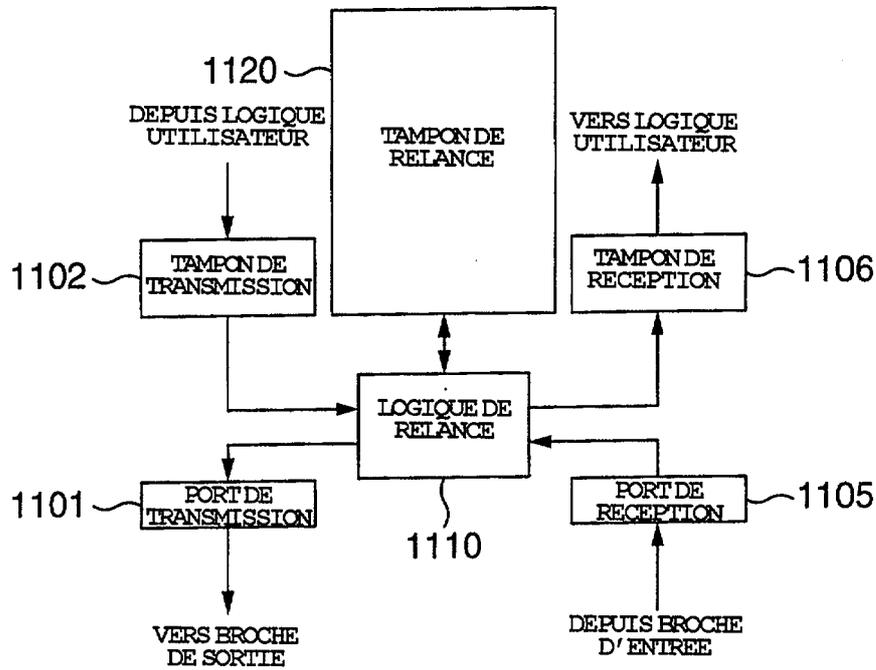
420), un champ d'adresse (521) pour désigner un contrôleur DMAC de lancement (120, 220, 320, 420) et un champ de séquence DMA (524) pour gérer une séquence de transfert lorsqu'un transfert DMA est distribué à une pluralité de paquets (500).

FIGURE 1



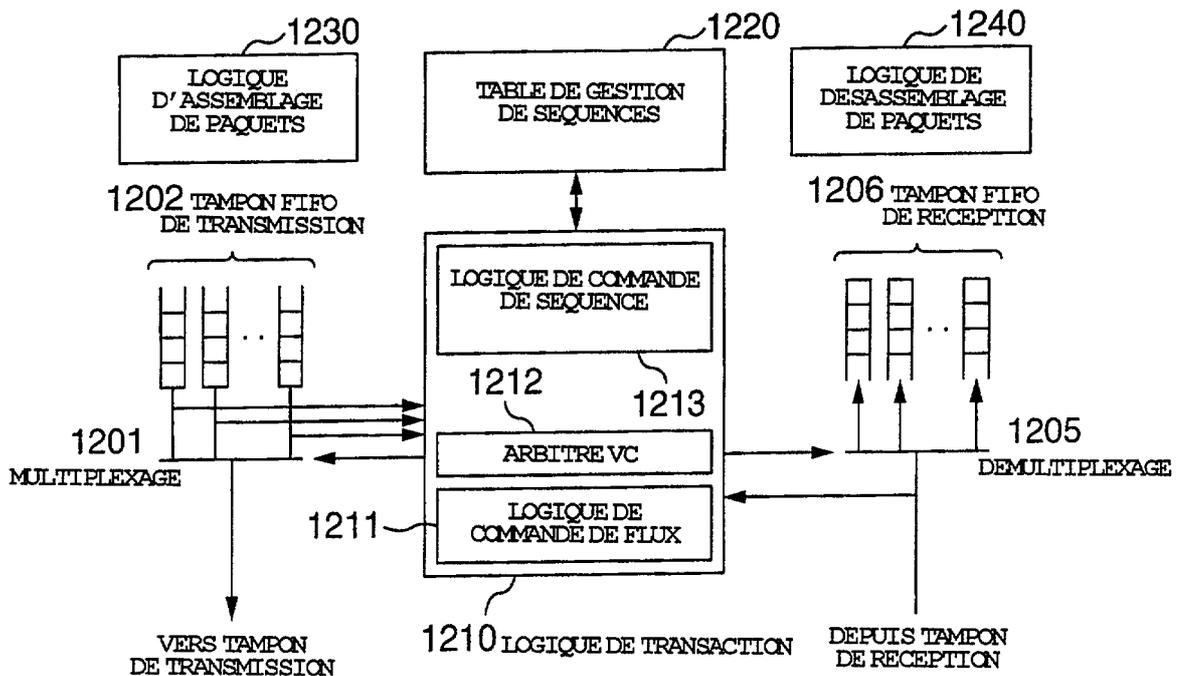
2/16

**FIGURE 2**



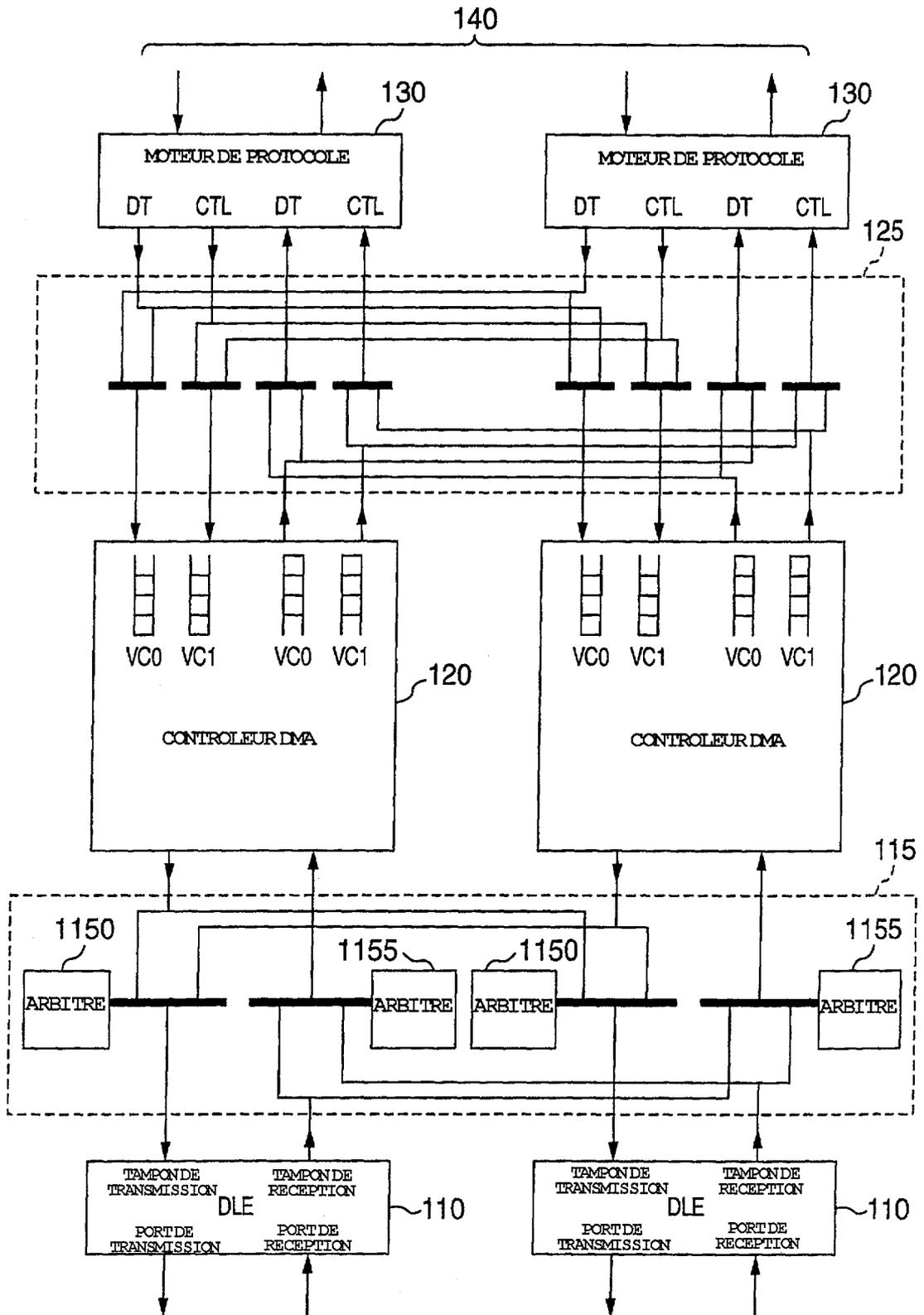
STRUCTURE DE MOTEUR DLE UTILISEE PAR LA PRESENTE INVENTION

**FIGURE 3**



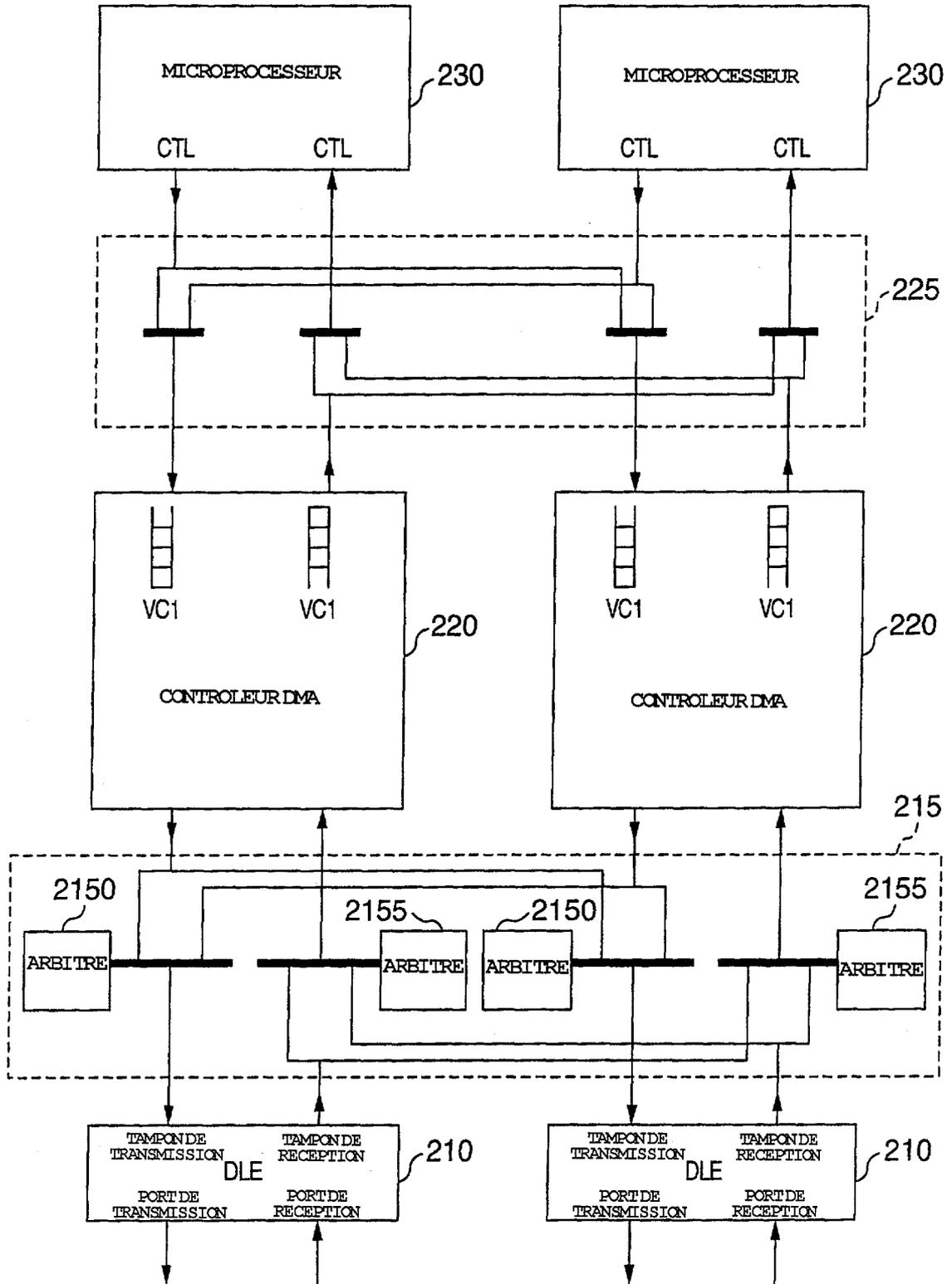
STRUCTURE DE CONTROLEUR DMAC UTILISEE PAR LA PRESENTE INVENTION

FIGURE 4



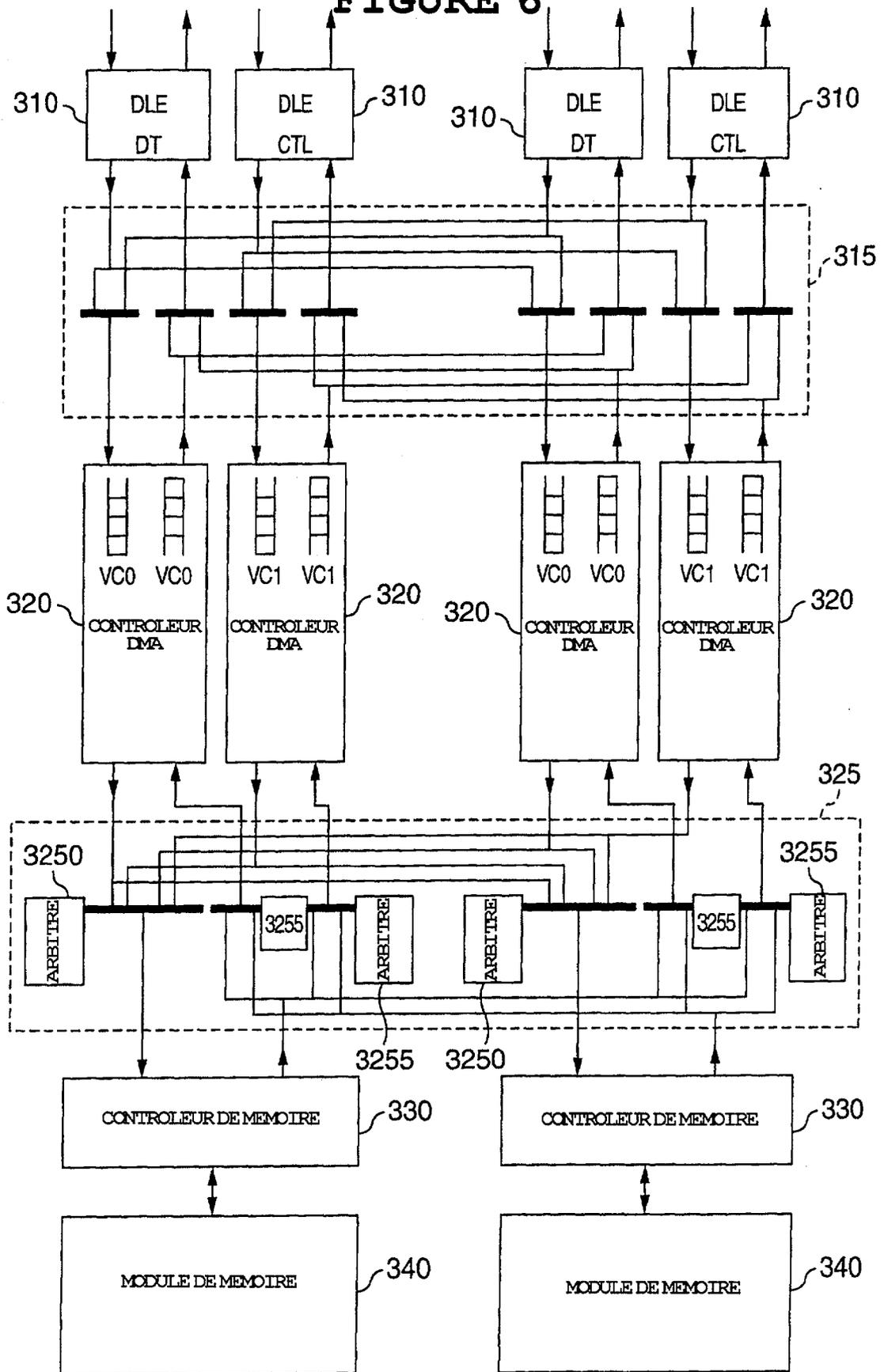
STRUCTURE D'ADAPTEUR DE CANAL UTILISEE PAR LA PRESENTE INVENTION

FIGURE 5



STRUCTURE D'ADAPTATEUR DE PROCESSEUR UTILISEE PAR LA PRESENTE INVENTION

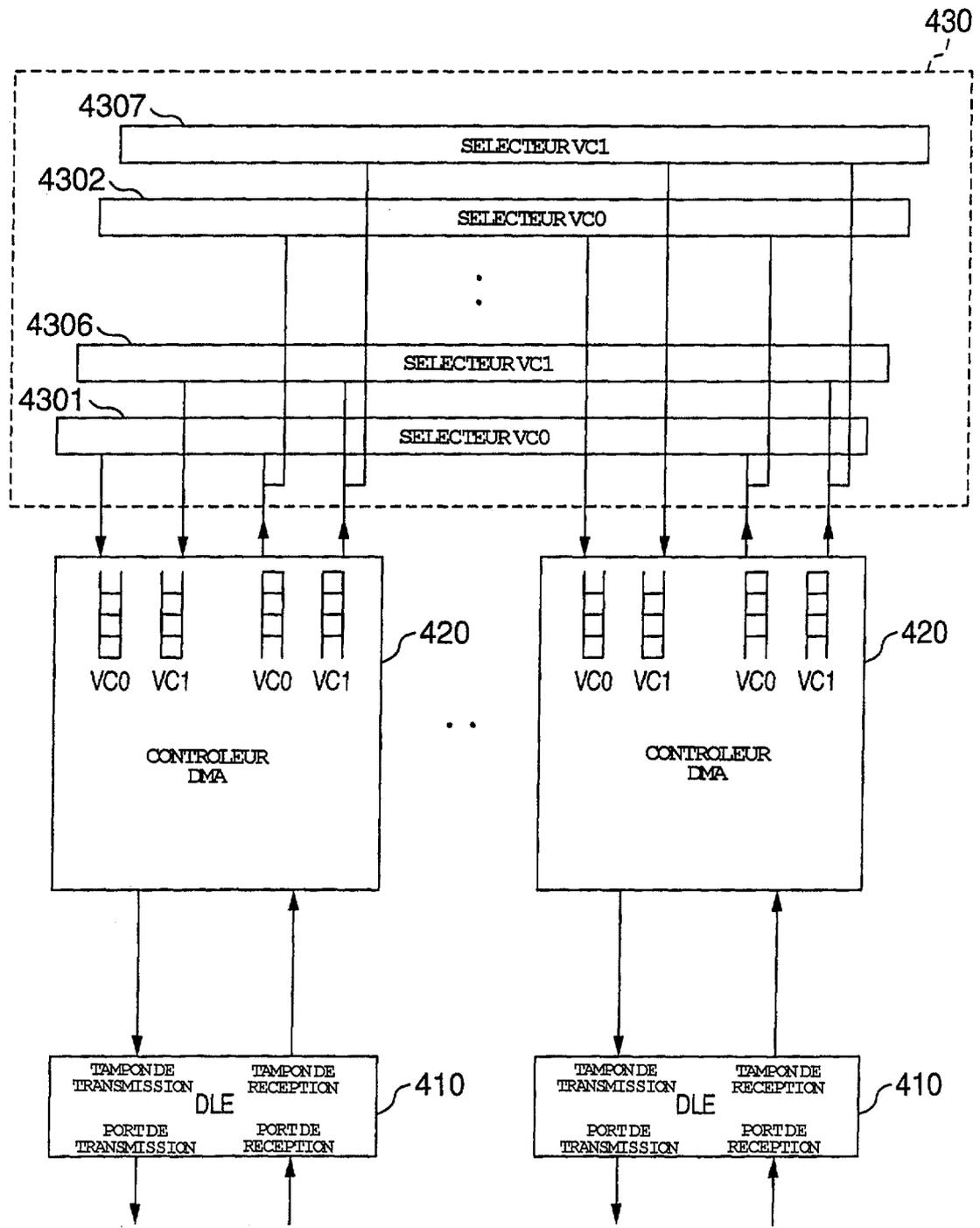
FIGURE 6



STRUCTURE D' ADAPTATEUR DE MEMOIRE UTILISEE PAR LA PRESENTE INVENTION

6/16

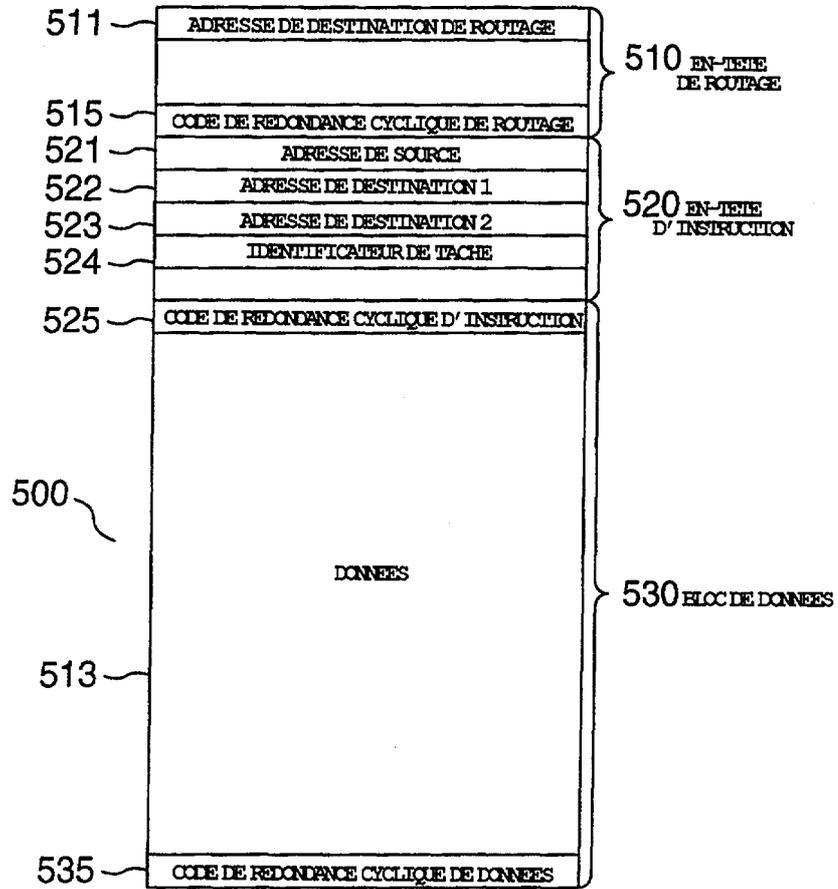
## FIGURE 7



STRUCTURE D'ADAPTEUR DE COMMUTATION UTILISEE PAR LA PRESENTE INVENTION

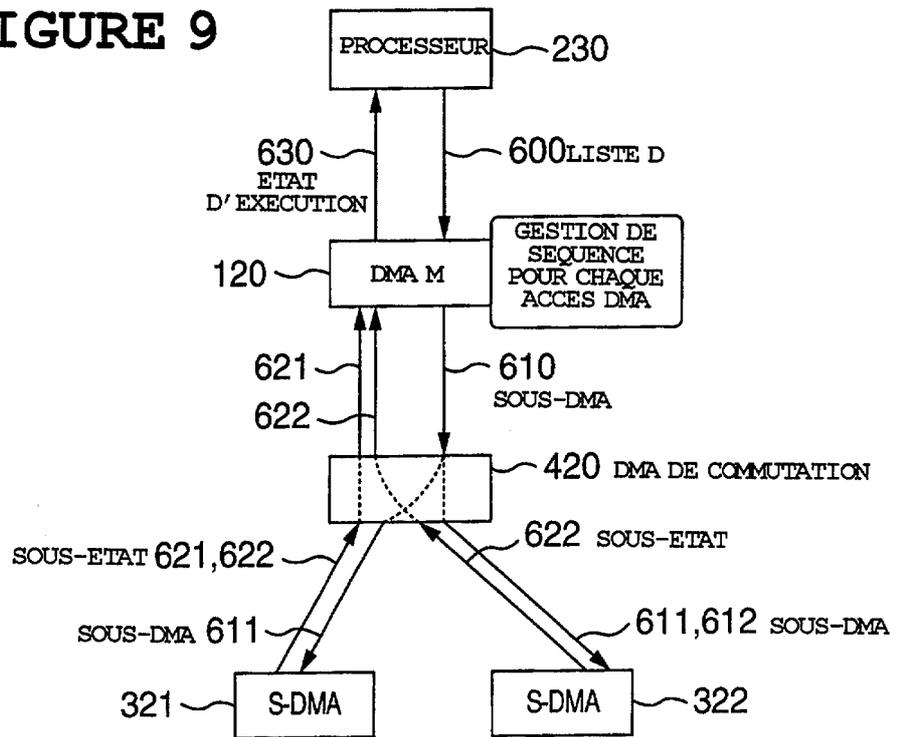
7/16

## FIGURE 8



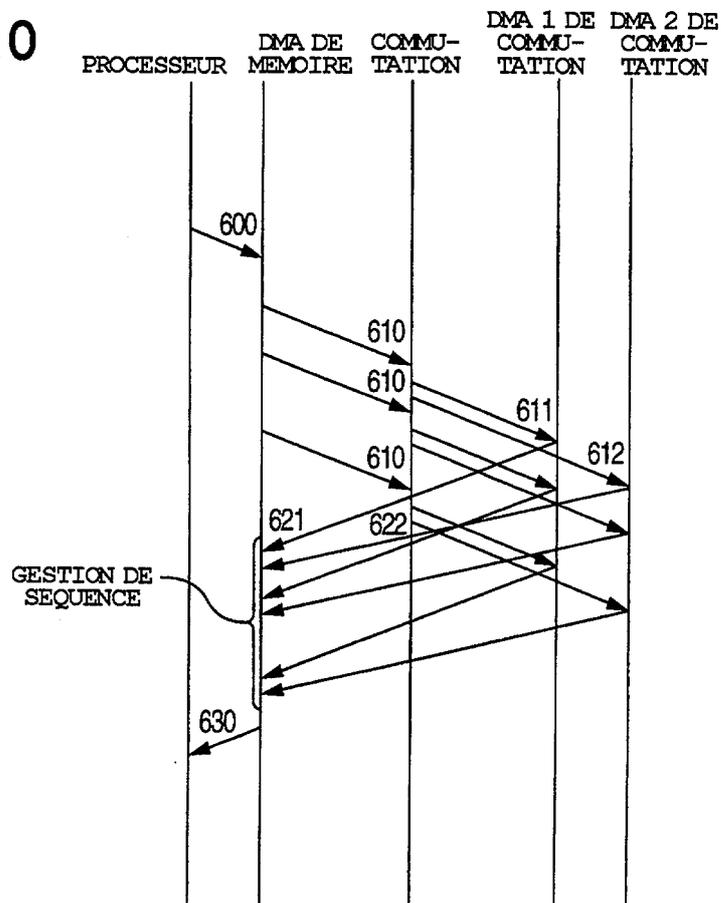
STRUCTURE DE PAQUET UTILISEE PAR LA PRESENTE INVENTION

FIGURE 9



FLUX DE PAQUET UTILISE PAR LA PRESENTE INVENTION

FIGURE 10

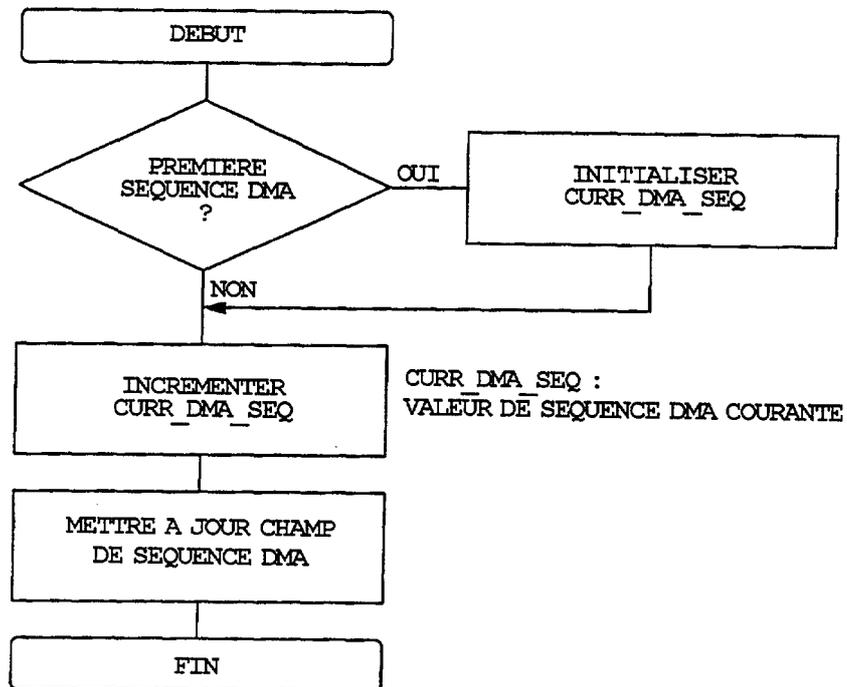


PROTOCOLE UTILISE PAR LA PRESENTE INVENTION



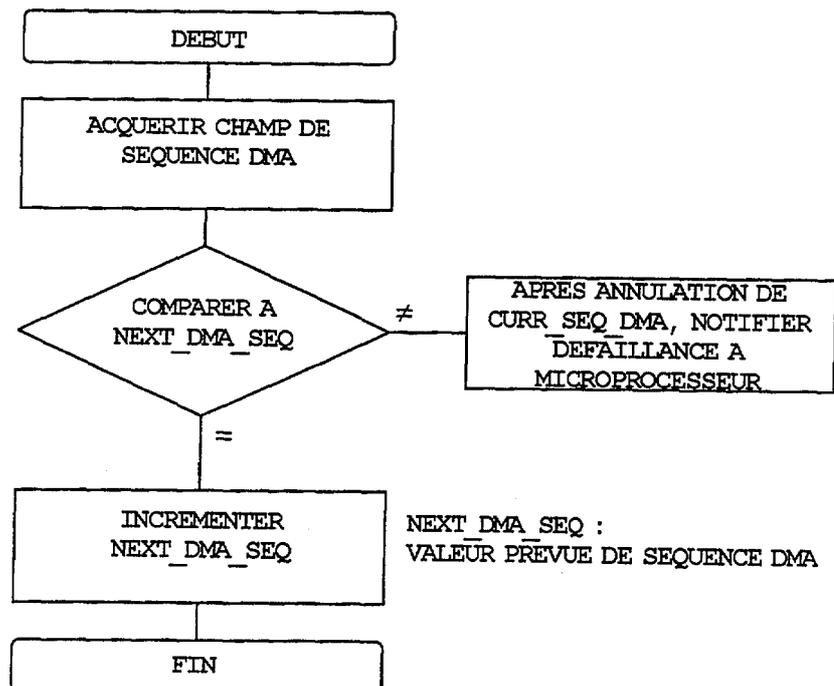
10/16

FIGURE 12



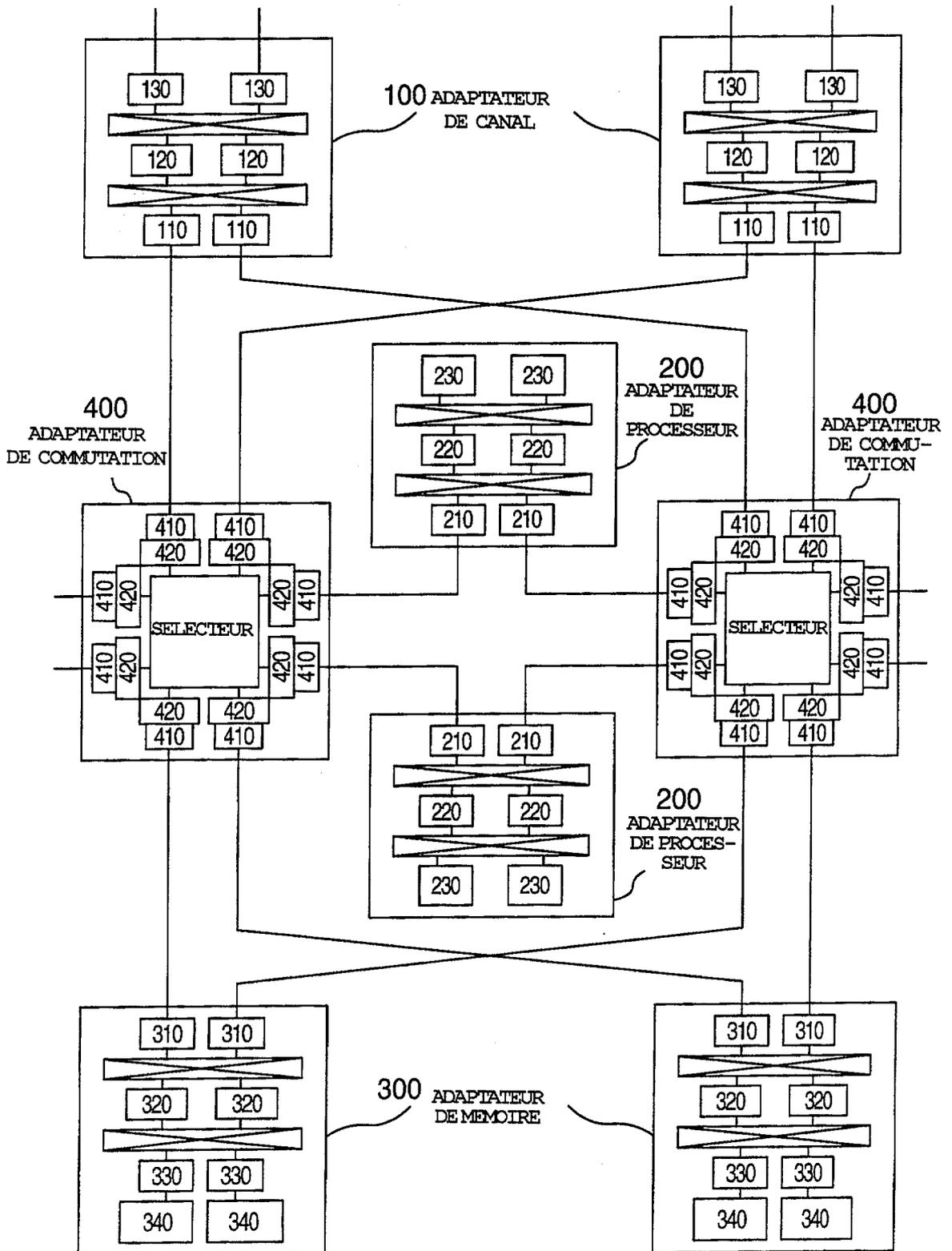
FLUX DE MISE A JOUR DE CHAMP DE SEQUENCE DMA  
DURANT UNE TRANSMISSION DE SOUS-DMA

FIGURE 13



FLUX DE CONFIRMATION DE CHAMP DE SEQUENCE  
DMA DURANT RECEPTION DE SOUS-ETAT DMA

FIGURE 14



CONTROLEUR DE DISQUE DE LA PRESENTE INVENTION

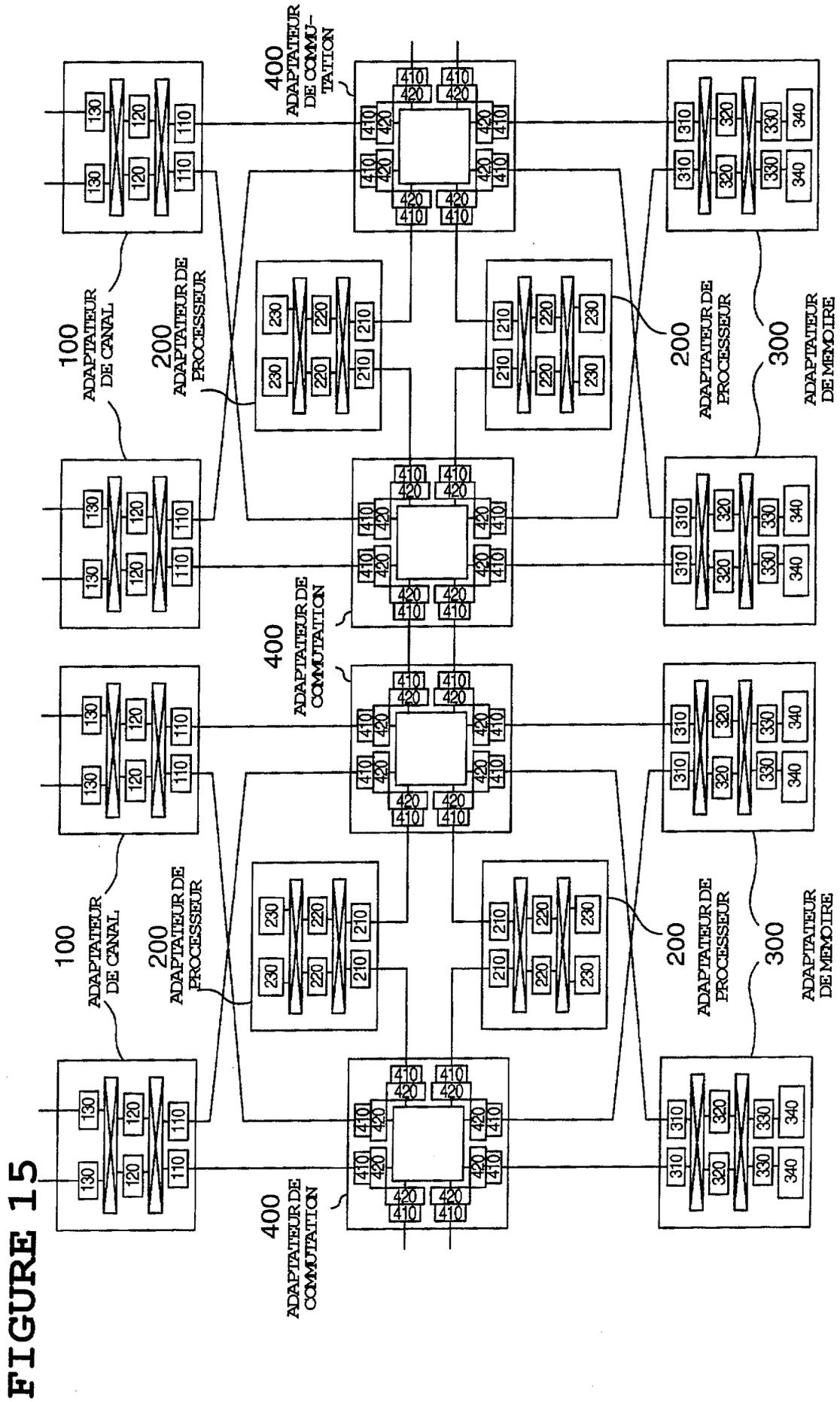
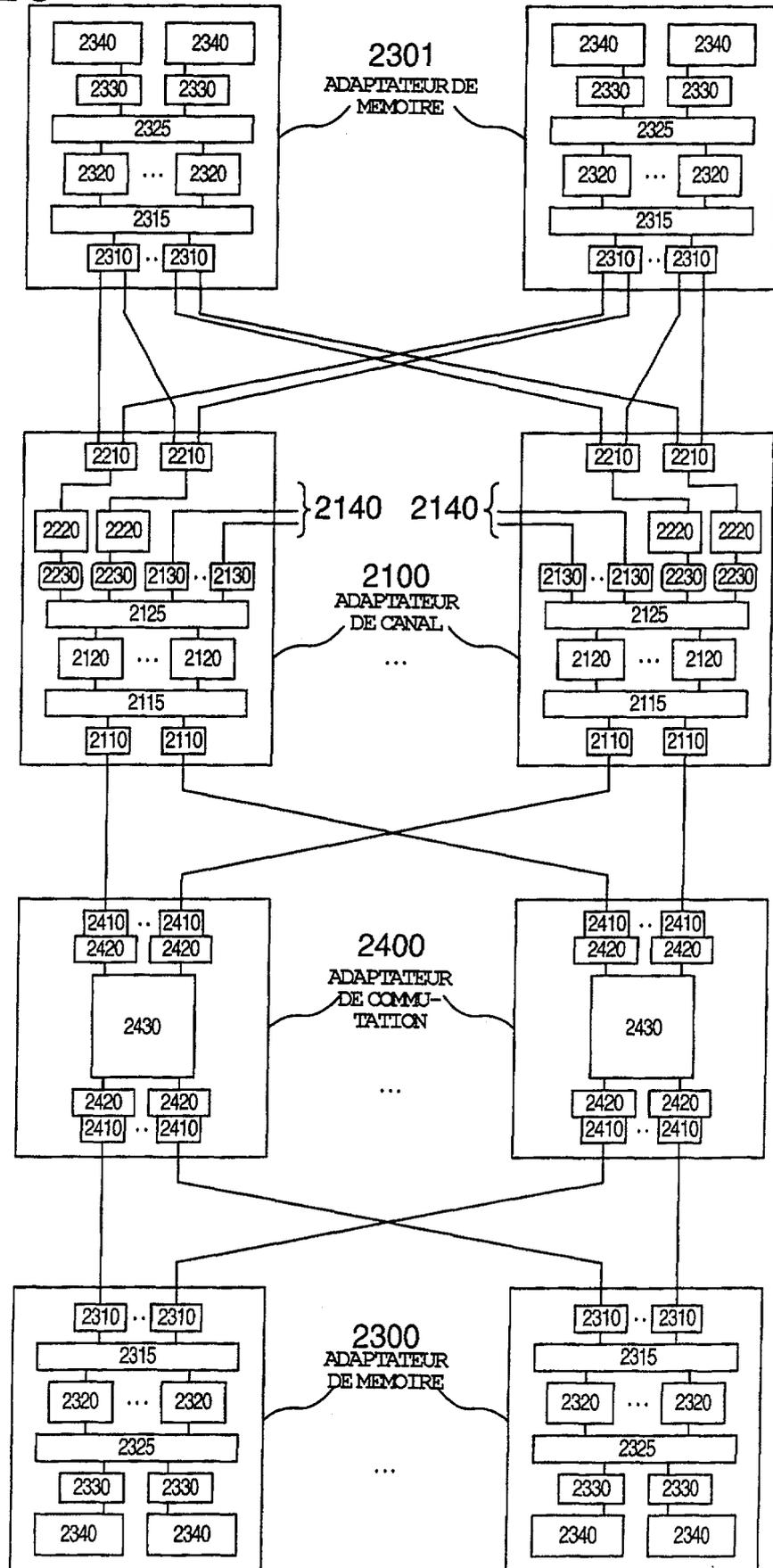


FIGURE 15

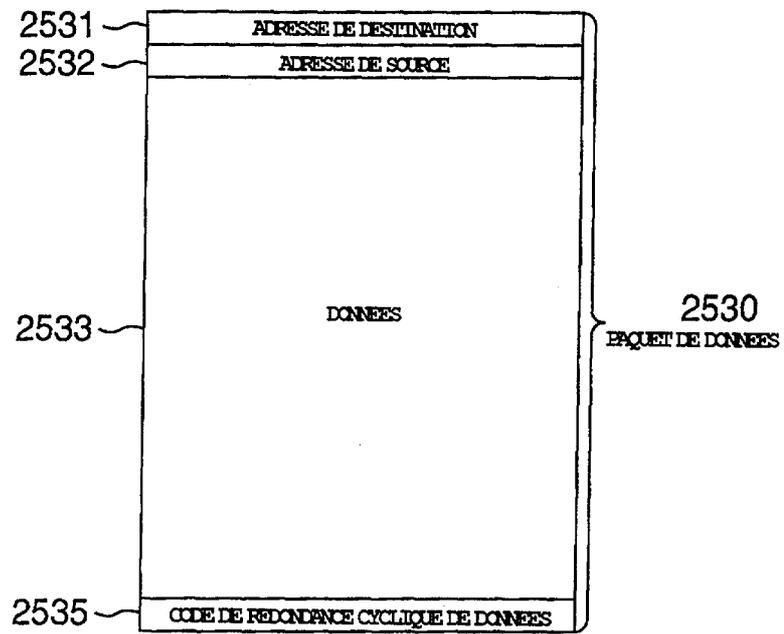
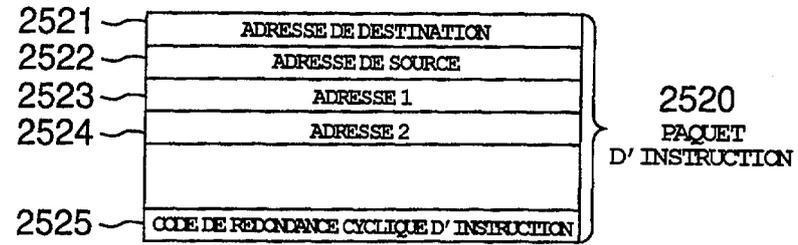
CONTROLEUR DE DISQUE DE LA PRESENTE INVENTION

FIGURE 16



CONFIGURATION DE RESEAU DE CONTROLEUR DE DISQUE CLASSIQUE

**FIGURE 17**



STRUCTURE DE PAQUET UTILISEE PAR LE SYSTEME DE COMMUNICATION DE TYPE CONNEKION

FIGURE 18

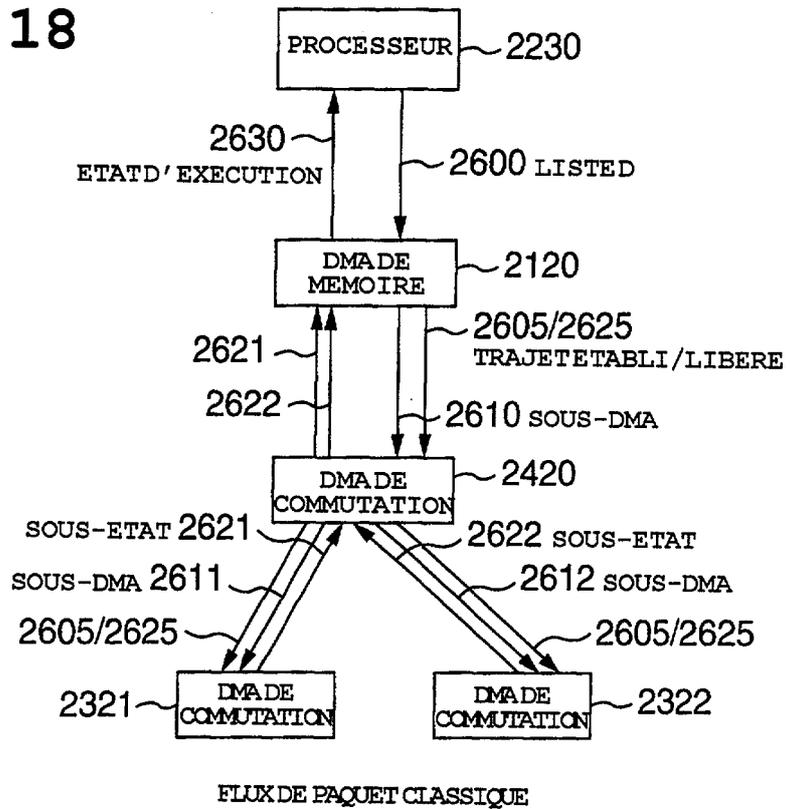
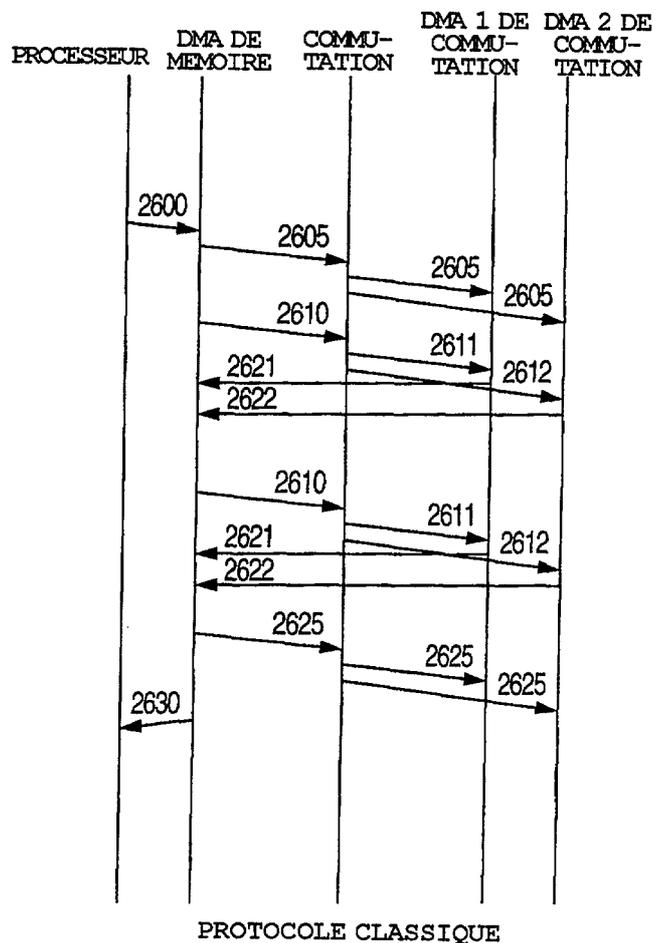
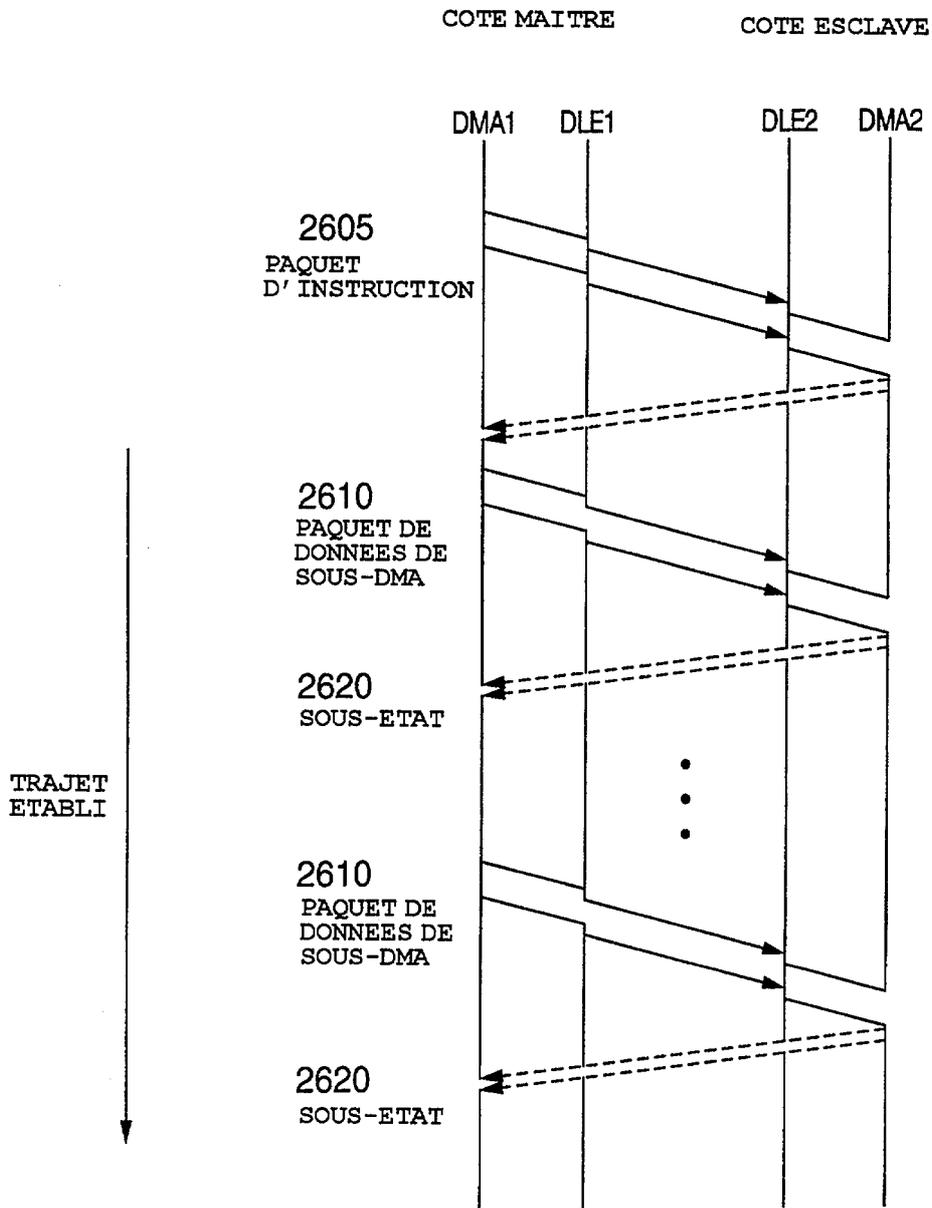


FIGURE 19



16/16

FIGURE 20



PROTOCOLE DE TRANSFERT DE COMMUNICATION NON-MULTIPLEX