RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 479 532**
(A n'utiliser que pour les commandes de reproduction).

A1

DEMANDE DE BREVET D'INVENTION

(21) N° **80 07288**

(54) **Procédé et dispositif pour gérer les transferts d'informations entre un ensemble mémoire et les différentes unités de traitement d'un système de traitement numérique de l'information.**

(51) Classification internationale (Int. Cl.³). **G 11 C 9/00; G 06 F 13/00.**

(22) Date de dépôt ..... 1er avril 1980.
(33) (32) (31) Priorité revendiquée :

(41) Date de la mise à la disposition du
public de la demande ..... B.O.P.I. — « Listes » n° 40 du 2-10-1981.

(71) Déposant : COMPAGNIE INTERNATIONALE POUR L'INFORMATIQUE CII-HONEYWELL BULL, résidant en France.

(72) Invention de : Michel Isert et Michel Jean-Marie Guillemet.

(73) Titulaire : *Idem* (71)

(74) Mandataire : Hervé Denis,
94, av. Gambetta, 75020 Paris.

La présente invention concerne généralement un procédé et un dispositif permettant de donner à un microprogrammeur chargé d'écrire des microprogrammes pour l'éxécution d'une tâche par un processeur d'un système de traitement numérique de l'information, une image particulière de l'ensemble mémoire du système, à savoir une image alignée à l'octet et de longueur variable sans se soucier notamment des problèmes de cadrage des informations et des temps de réponse de l'ensemble mémoire. Plus particulièrement, la présente invention a pour objet un procédé et un dispositif pour gérer des transferts d'informations entre un ensemble mémoire et les différentes unités de traitement du système au cours de l'éxécution d'une tâche par le ou un des processeurs du système tout en minimisant le nombre d'accès mémoire.

De façon générale, les informations à lire ou à écrire dans l'ensemble mémoire d'un système numérique de traitement de l'information, au cours de l'exécution d'une tâche par un processeur du système, sont essentiellement des chaînes d'octets de longueur quelconque et situées à une adresse quelconque avec progression de l'adresse à droite ou à gauche. Les transferts des informations concernant aussi bien des instructions que des données, sont sous le contrôle d'un ensemble de microprogrammes pré-enregistrés. Ces microprogrammes sont sélectivement appelés et exécutés au cours de l'exécution d'une tâche.

Le microprogrammeur chargé d'écrire ces microprogrammes doit tenir compte de la structure des informations, de leur mode d'enregistrement dans l'ensemble mémoire, de leur longueur, de leur cadrage,... . En outre, les informations relatives à des instructions et à des données sont transférées au fur et à mesure des besoins.

L'un des buts visés par l'invention est de simplifier

l'écriture de ces microprogrammes en dégageant au maximum le microprogrameur des contraintes imposées par les identifications des informations à traiter. Dans ces conditions, les contraintes supprimées au niveau de l'écriture des microprogrammes, sont prises en compte et gérées automatiquement par le système conforme à l'invention.

Un autre but visé par l'invention est de gérer les transferts des informations aussi bien au niveau des instructions qu'au niveau des données, de façon à limiter au maximum les accès mémoire et gagner ainsi du temps sur l'exécution d'une tâche, cette gestion des transferts étant assurée automatiquement par le système conforme à l'invention.

Un autre but visé par l'invention est de se dégager des contraintes imposées par les temps de réponse variables de l'ensemble mémoire, temps de réponse qui dépendent de la configuration du système.

L'invention propose donc un procédé pour gérer des transferts d'informations entre un ensemble mémoire et différentes unités de traitement au cours de l'exécution d'une tâche par un système de traitement numérique de l'information comprenant au moins un processeur associé à un cache, une unité de mémoire principale, le cache et l'unité de mémoire principale formant ledit ensemble mémoire, une unité d'entrée-sortie, et un sous-système incluant lesdites unités de traitement reliées entre elles par au moins un bus de liaison, caractérisé en ce qu'il consiste :

- à interposer à l'intérieur du sous-système, entre l'ensemble mémoire et les unités de traitement du système, une unité d'interface comprenant un ensemble de mémoires tampon dits tampons par lesquels transitent aussi bien des informations relatives à des instructions d'un programme à

exécuter et à transférer vers lesdites unités, que des informations relatives à des données demandées à l'ensemble mémoire par les unités de traitement ou à écrire dans l'ensemble mémoire,

- à effectuer des transferts automatiques d'informations relatives à des instructions entre l'ensemble mémoire et ladite unité d'interface, en avance par rapport à l'exécution proprement dite de ces instructions,

- à effectuer des transferts automatiques d'informations relatives à des données entre l'ensemble mémoire et ladite unité d'interface, en avance par rapport à leur utilisation par les unités de traitement, et

- à effectuer des transferts automatiques d'informations relatives à des données entre les unités de traitement et l'ensemble mémoire,

ces transferts automatiques étant indépendants des microprogrammes chargés d'exécuter les instructions relatives au traitement d'une tâche.

Selon une autre caractéristique de l'invention, le procédé consiste à donner au microprogrammeur chargé d'écrire les microprogrammes d'exécution des instructions d'un programme, une image de l'ensemble mémoire aligné à l'octet et à longueur variable, et à gérer automatiquement, par l'intermédiaire de microfonctions, tous les problèmes relatifs notamment au cadrage et à la longueur des informations à transférer.

Selon une autre caractéristique de l'invention, le procédé consiste à gérer le contenu des différents tampons de l'unité d'interface par l'intermédiaire de compteurs et de pointeurs, en fonction de la disponibilité pour ces tampons

à recevoir et à transmettre des informations entre l'ensemble mémoire et les unités de traitement du système.

Selon une autre caractéristique de l'invention, le procédé consiste à faire émettre automatiquement, par ladite unité d'interface, des requêtes de transferts automatiques d'informations relatives à des instructions ou à des données depuis l'ensemble mémoire vers au moins un tampon dit tampon de lecture de l'unité d'interface, dès que ce tampon a suffisamment de place disponible pour recevoir au moins une partie de ces informations.

Selon une autre caractéristique de l'invention, le procédé consiste à faire émettre automatiquement, par ladite unité d'interface, des requêtes de transferts automatiques d'informations relatives à des données entre au moins un tampon dit tampon d'écriture de l'unité d'interface et ledit ensemble mémoire, dès que ce tampon contient au moins une partie de ces données.

Selon une autre caractéristique de l'invention, le procédé consiste à mettre automatiquement en attente l'exécution d'un microprogramme, si les informations demandées par ce microprogramme ne sont pas encore disponsibles dans le tampon associé de ladite unité d'interface.

Selon une autre caractéristique de l'invention, le procédé consiste à transférer automatiquement des chaînes d'octets entre l'ensemble mémoire et des unités de traitement sans que le microprogrammeur connaisse au départ la longueur de ces chaînes, la fin d'une chaîne étant automatiquement détectée au niveau des tampons associés de ladite unité d'interface et par lesquelles transitent ces chaînes d'octets.

Selon une autre caractéristique de l'invention, le procédé

consiste, lorsque plusieurs tampons de ladite unité d'interface sont émetteurs d'une requête de transfert d'informations, à gérer la priorité de ces requêtes par des circuits de priorité.

Selon une autre caractéristique de l'invention, le procédé consiste à interposer notamment un circuit d'alignement des informations entre le tampon de lecture de ladite unité d'interface et le bus de liaison reliant les différentes unités de traitement du sous-système, et à interposer également un circuit d'alignement des informations entre ce même bus de liaison et le tampon d'écriture de ladite unité d'interface, ces circuits d'alignement contrôlant automatiquement le cadrage des informations donné par les premiers bits de poids faible des adresses mémoire de ces informations.

L'invention propose également un dispositif permettant la mise en oeuvre du procédé, dispositif qui est essentiellement constitué par une unité d'interface comprenant un tampon programme par lequel transitent les informations relatives aux instructions d'un programme à exécuter, au moins un tampon de lecture par lequel transitent les informations transmises depuis l'ensemble mémoire du système vers les unités de traitement du système, au moins un tampon d'écriture par lequel transitent les informations relatives à des données provenant des unités de traitement du système et à écrire dans l'ensemble mémoire du système, les tampons de lecture et d'écriture étant reliés d'une part à l'ensemble mémoire du système et au bus de liaison reliant les différentes unités de traitement, le tampon programme étant relié à l'ensemble mémoire du système et à certaines des unités de traitement du système par des lignes de liaison unidirectionnelles.

Selon une autre caractéristique du dispositif conforme à

l'invention, l'évolution des contenus des tampons précités est gérée automatiquement par des compteurs et par des pointeurs mis à jour par des microfonctions.

Selon une autre caractéristique du dispositif conforme à l'invention, des circuits d'alignement des informations sont associés aux tampons de ladite unité d'interface, pour gérer automatiquement le cadrage de ces informations.

Selon un avantage important de l'invention, le travail du microprogrammeur chargé d'écrire des microprogrammes d'exécution d'instructions, est considérablement simplifié du fait qu'il n'a pas à se soucier notamment des problèmes de cadrage des informations et des temps de réponse de l'ensemble mémoire. En effet, tous ces problèmes sont pris en compte par l'unité d'interface précitée qui assure en outre une parfaite synchronisation entre l'exécution des microprogrammes et les transferts des informations entre l'ensemble mémoire et les unités de traitement du système.

L'invention va être décrite plus en détail en référence aux dessins annexés donnant à titre d'illustration un exemple de réalisation de l'invention.

La figure 1 représente de façon schématique un système de traitement de l'information conforme à l'invention.

La figure 2 représente d'une façon simplifiée les principaux éléments constitutifs de l'unité d'interface DIM représentée sur la figure 1.

La figure 3 représente le tampon programme de l'unité DIM dans le cas d'un transfert d'informations relatives à une instruction d'un programme à exécuter.

La figure 4 représente un tableau indiquant le mode de

transfert des différents octets de l'instruction précitée à exécuter, entre le tampon programme et l'unité de traitement du système chargée du décodage de cette instruction.

La figure 5 représente une chaîne d'octets enregistrés dans le cache du processeur du système, et qui doit être transférée dans l'une des unités de traitement du système.

La figure 6 représente sous forme d'un diagramme les différentes phases du transfert de la chaîne d'octets représentée sur la figure 5.

Les figures 7 et 8 représentant un autre exemple de transfert d'une chaîne d'octets entre le cache du processeur et l'une des unités de traitement du système.

La figure 9 représente une chaîne d'octets qui doit être transférée depuis l'une des unités de traitement du système vers l'ensemble mémoire du système.

La figure 10 représente sous forme d'un diagramme les différentes phases du transfert de la chaîne d'octets représentée sur la figure 9.

Avant de décrire l'invention en référence aux dessins, il est à noter que les différentes unités, registres, signaux, tampons utilisés par le système qui sera décrit ci-après seront désignés par les abréviations des expressions anglo-saxonnes couramment utilisées et ce, pour une meilleure compréhension par l'homme du métier. Une liste de ces abréviations est donnée ci-dessous avec leur signification en langue anglaise :

ACM : Adress Computation Machine
ALM : Arithmetic and Logic Machine
CSS : Central Sub - System

CSU : Control Store Unit
DIM : Data and Instruction Machine
IOC : Input - Output Controller
MMU : Main Memory Unit
PIM : Pilot Machine
SCM : Scientific Computation Machine
BP : BUSA Pointer
BC : BUSA Counter
MP : Memory Pointer
AMP : Advanced Memory Pointer
AMC : Advanced Memory Counter
DIRQ : DIM Request
ACRQ : ACM Request
RQA : Request Accepted
DTA : Data Available
BP-DT: Bus Processor - Data
WB : Write Buffer
RB0 : Read Buffer N°. 0
RB1 : Read Buffer N°. 1
PB : Program Buffer
RT : Read Trap Buffer
WT : Write Trap Buffer

Sur la figure 1 est représenté un système de traitement numérique de l'information comprenant au moins :

- une unité de mémoire principale MMU,
- un processeur P auquel est associé un cache C ou mémoire de faible capacité à accès rapide, et
- une unité d'entrée-sortie IOC.

Le processeur P et son cache C font partie d'un sous-système CSS comprenant plusieurs unités parmi lesquelles :

- une unité DIM chargée d'effectuer les différents accès mémoire soit dans le cache C, soit dans l'unité de mémoire

principale MMU,

- une unité ACM chargée d'effectuer les différents calculs d'adresse des informations à lire dans l'ensemble mémoire du système comprenant l'unité de mémoire principale MMU et le cache C du processeur P,

- une unité ALM chargée d'effectuer les différentes opérations arithmétiques et logiques,

- une unité PIM chargée de piloter les différents branchements entre les différentes unités du sous-système CSS,

- une unité CSU où sont enregistrés les différents microprogrammes permettant d'assurer l'exécution d'une tâche par le sous-système CSS, et

- une unité SCM chargée d'effectuer notamment les opérations arithmétiques en virgule flottante.

Le sous-système CSS est relié par un bus principal de liaison UMI avec la ou les unités de mémoire principale MMU et la ou les unités d'entrée-sortie IOC.

Le cache C du processeur P communique avec l'unité DIM par un bus de liaison BP-DT, et avec l'unité ACM par des lignes unidirectionnelles AC-AD pour le transfert des adresses entre l'unité ACM et le cache C.

Les unités ACM, DIM, ALM, PIM et SCM communiquent entre elles par un bus de liaison BUSA. Des lignes unidirectionnelles de liaisons spécifiques INST assurent des liaisons particulières à partir de l'unité DIM en direction des unités PIM, ACM et ALM.

L'invention se situe essentiellement au niveau de l'unité DIM par laquelle transitent toutes les informations relatives à des instructions et à des données demandées ou envoyées par le processeur P au cours de l'exécution d'une tâche.

L'unité DIM est essentiellement constituée de mémoires tampon dits tampons. Les contenus et l'évolution des contenus de ces tampons sont gérés par des micro-programmes enregistrés dans l'unité CSU du sous-système CSS, comme cela sera explicité plus loin.

En se reportant à la figure 2, il a été représenté de façon schématique une unité DIM faisant l'objet de la présente invention.

Cette unité DIM comprend notamment :

- un tampon programme PB de 16 octets ou bytes, par lequel transitent les informations relatives aux instructions à exécuter;

- deux tampons de lecture RB0 et RB1 de 16 octets chacun, par lesquels transitent les informations relatives aux données provenant de l'ensemble mémoire du système,

- un tampon d'écriture WB de 16 octets, par lequel transitent les informations relatives aux données à écrire dans l'ensemble mémoire du système,

- un tampon de lecture RT de 8 octets pour traiter les interruptions de séquence, et

- un tampon d'écriture WT de 8 octets pour traiter également les interruptions de séquence.

Le cache C du processeur P est relié par l'intermédiaire du bus de liaison BP-BT aux entrées d'un premier multiplexeur M1. Les sorties de ce multiplexeur M1 sont reliées par des lignes unidirectionnelles B1 aux entrées du tampon programme PB et aux entrées des tampons de lecture RB0, RB1 et RT.

En sortie, les tampons de lecture sont reliés à un second multiplexeur M2. Les sorties de ce multiplexeur M2 sont reliées aux entrées d'un troisième multiplexeur M3 par l'intermédiaire de quatre conducteurs C1 à C4. Les sorties du multiplexeur M3 sont reliées aux entrées d'un premier circuit d'alignement des informations relatives aux données, par l'intermédiaire de quatre conducteurs C5 à C8. Les sorties du circuit d'alignement CA1 sont reliées aux entrées d'un quatrième multiplexeur M4, par l'intermédiaire de quatre conducteurs C9 à C12. Les sorties du multiplexeur M4 sont reliées par le bus BUSA aux différentes unités du sous-système CSS.

En sortie, le tampon programme PB est relié :

- aux entrées du second multiplexeur M2,

- aux entrées d'un cinquième multiplexeur M5, par l'intermédiaire des deux conducteurs C13 et C14,

- aux entrées d'un second circuit d'alignement CA2 des informations relatives aux instructions, par l'intermédiaire de quatre conducteurs : les conducteurs C15 et C16 précités, et deux conducteurs C17 et C18 prévus en sortie du multiplexeur M5. Les sorties du second circuit d'alignement CA2 sont reliées par les lignes unidirectionnelles de liaison INST aux unités PIM, ACM et ALM ; les liaisons B1 précitées en sortie du premier multiplexeur M1 étant également reliées aux entrées du multiplexeur M3 et aux entrées du multiplexeur M5.

Le bus de liaison BUSA est relié à un troisième circuit d'alignement CA3 des informations relatives à des données à écrire dans l'ensemble mémoire du système. Les sorties du troisième circuit d'alignement CA3 sont reliées par quatre conducteurs C5 à C8 aux entrées des tampons d'écriture WT et WB. En sortie, ces tampons sont reliés par le bus de liaison BP-DT à l'ensemble mémoire du système.

Avant de décrire le fonctionnement de cette unité DIM et plus précisément ses fonctions au cours de l'exécution d'une tâche par le proceseur P, il est important de définir au préalable la structure des informations qui vont transiter par cette unité.

Dans le cache C, qui peut être à plusieurs niveaux, les informations sont enregistrées par blocs de quatre mots chacun, un mot contenant quatre octets.

Les transferts entre le cache C et l'unité DIM s'effectuent par mot ou double mot, et les transferts entre l'unité DIM et le bus de liaison BUSA concernent de un à quatre octets. Les lignes spécifiques INST peuvent transmettre des données sur deux ou quatre octets.

Supposons le processeur P amené à exécuter une tâche. L'exécution de cette tâche va donner lieu au déroulement d'un programme fourni par un utilisateur et à l'appel de microprogrammes pré-enregistrés chargés d'exécuter les instructions de ce programme avec les transferts adéquates d'informations entre les unités de traitement du processeur P et l'ensemble mémoire du système.

Le programme, que l'on suppose enregistré en tout ou partie dans l'unité de mémoire principale MMU, est initialisé par

le système, puis exécuté en tenant compte de ruptures de séquence éventuelles.

Après l'initialisation du programme par le système, le premier tampon concerné de l'unité DIM est le tampon programme PB. En effet, la première opération à effectuer par le sous-système CSS est le décodage de la première instruction du programme. Ce décodage est effectué par l'unité PIM du sous-système CSS.

Avant de décrire la façon dont les informations sont envoyées à l'unité PIM pour qu'elle effectue le décodage de l'instruction, il va être décrit ci-après les éléments principaux qui assurent le contrôle du tampon programme PB. Ces éléments sont d'une part des microfonctions, et d'autre part des compteurs mémoire et des pointeurs.

Ces microfonctions, au nombre de 4, sont les suivantes :

1°) IRP : cette microfonction permet d'initialiser les transferts des informations relatives aux instructions entre l'ensemble mémoire du système et le tampon programme PB.

L'adresse de la première instruction du programme à lire est donnée par l'unité ACM et les trois bits de poids faible de cette même adresse donnent le cadrage des informations relatives aux instructions à l'intérieur du tampon programme PB.

2°) SIN : cette microfonction permet d'extraire les octets des informations relatives aux instructions présentes dans le buffer programme PB, et de les envoyer sur les lignes unidirectionnelles INST en direction de l'unité PIM, en ce qui concerne notamment le code opération de l'instruction pour calculer l'adresse de la première microinstruction du microprogramme chargé de l'interprétation de l'instruction.

3°) EPR : cette microfonction permet d'extraire les octets du tampon programme PB pour les envoyer sur le bus BUSA, à partir du quatrième octet de l'instruction dont le code opération est contenu dans le premier octet transmis à l'unité PIM par les lignes unidirectionnelles INST.

4°) SIF : cette microfonction est chargée d'émettre des requêtes en vue d'alimenter le tampon programme PB avec les octets de l'instruction suivante du programme, à condition qu'il y ait assez de place disponible dans le tampon programme PB.

Comme indiqué précédemment, le contenu du tampon programme PB est également sous le contrôle d'un compteur mémoire AMC et de trois pointeurs MP, AMP, BP.

Le compteur mémoire AMC contient le nombre d'octets (moins un) à échanger avec l'ensemble mémoire du système à la fin de la requête de lecture préalablement initialisée. Ce compteur mémoire est sur 5 bits de long, et le bit de poids fort de ce compteur signifie que le contenu du compteur est vide lorsque ce bit est à 1.

Le pointeur MP pointe sur le prochain octet à échanger avec l'ensemble mémoire du système. Ce pointeur est sur 5 bits de long.

Le pointeur BP pointe sur le prochain octet à échanger avec le bus BUSA. Ce pointeur est sur 5 bits de long.

Bien évidemment, le contenu du compteur mémoire AMC et des pointeurs MP, AMP et BP sont mis à jour régulièrement et automatiquement comme cela ressortira clairement plus loin.

Il est possible maintenant de revenir au décodage de la

première instruction du programme à exécuter.

Pour cela, la première microfonction précitée IRP est exécutée afin d'initialiser le transfert des octets de l'instruction entre l'ensemble mémoire du système et le tampon programme PB. L'adresse de cette première instruction est donnée par l'unité ACM et les trois bits de poids faible de cette adresse donnent le cadrage des octets de l'instruction à l'intérieur du tampon programme PB. La microfonction IRP permet de lire un double-mot dans le cache C du processeur P ou dans l'ensemble mémoire MMU du système. Ce double-mot est envoyé dans le tampon programme PB, par exemple dans les huit premiers octets de ce dernier. En se référant à la figure 3, supposons que les quatre premiers octets de l'instruction contenant respectivement les informations A, B, C et D, se retrouvent respectivement dans les octets 2, 3, 4 et 5 du tampon programme PB. Dans ces conditions, l'information A ou octet A de l'information est décalée de 2 octets par rapport à l'octet 0 du tampon programme PB. Ce décadrage est donné par les trois bits de poids faible de l'adresse de l'instruction. Simultanément, les pointeurs MP, AMP et BP prennent les valeurs de ces poids faibles. Le compteur mémoire AMC associé au tampon programme BP prend la valeur qui correspond au nombre d'octets (moins un) à lire à partir de l'octet A de l'instruction pour arriver à une limite de double-mot dans le tampon programme PB. Dans l'exemple illustré sur la figure 3, il faut par conséquent lire six octets de manière à être cadré sur l'octet 8 du tampon programme PB.

Ensuite, il faut envoyer les premiers octets de l'instruction à l'unité PIM chargée de décoder le code opération de l'instruction, code opération qui se trouve dans le premier mot (et plus précisément dans le premier octet) de l'instruction. Ce transfert est effectué par la microfonction SIN qui va envoyer les quatre octets A, B, C

et D de l'instruction à l'unité PIM par l'intermédiaire des lignes unidirectionnelles INST.

Cependant, il faut tenir compte du cadrage de l'instruction dans le tampon programme PB, pour retrouver dans le bon ordre les octets A, B, C, D de l'instruction sur les lignes INST. En se reportant aux figures 3, 4, il est représenté le transfert des octets A, B, C et D, transfert qui est sous le contrôle du tableau représenté sur la figure 4. On suppose que le tampon programme PB se présente sous la forme d'un tableau de quatre lignes et de quatre colonnes, ces dernières étant numérotées de 0 à 3. On considère également les deux bits de poids faible du pointeur BP qui donnent le décalage du premier octet A de l'instruction à l'intérieur du mot contenant cet octet. Ce décalage peut donc prendre la valeur 0, 1, 2 ou 3 (valeurs indiquées en binaire sur la figure 4). Dans l'exemple présenté, l'octet A est décalé de 2 octets. Dans ces conditions, il faut se reporter à la ligne 3 du tableau de la figure 4 qui indique : le premier octet à transférer sur les lignes INST est l'octet de l'information qui se trouve dans la colonne 2 (c'est-à-dire l'octet A), le second octet à transférer est l'octet qui se trouve dans la colonne 3 du tampon programme PB (c'est-à-dire l'octet B), etc... Ce recadrage est effectué par le circuit d'alignement CA2 de la figure 2.

Il est important de noter qu'à ce stade, on ne connait pas la longueur de l'instruction. Par contre, l'unité PIM ayant reçu le code opération de l'instruction, va décoder celui-ci et permettre le branchement vers un microprogramme associé à cette instruction. Dans ce microprogramme spécifique à cette instruction, le microprogrammeur en écrivant le microprogramme connait la nature de l'instruction à traiter et par conséquent sa longueur. Les autres octets éventuels de l'instruction sont ensuite transmis aux unités concernées

du sous-système CSS par l'intermédiaire du bus de liaison BUSA sous la commande de la microfonction EPR.

Pour l'exécution de cette microfonction EPR, le microprogrammeur a juste à donner le nombre d'octets à lire sans se soucier du cadrage de ces octets à l'intérieur du tampon programme PB. Ces octets sont transférés par les multiplexeurs M2, M3, jusqu'au circuit d'alignement CA1, avant d'être envoyés sur le bus BUSA par l'intermédiaire du multiplexeur M4.

Selon une caractéristique importante de l'invention, des requêtes de lecture peuvent être émises automatiquement afin de pouvoir lire en avance au moins les premiers octets de l'instruction suivante du programme à exécuter, afin de placer ces octets en avance dans le tampon programme PB en vue de leur décodage par l'unité PIM.

Une telle requête automatique peut être demandée par la microfonction SIN, à condition toutefois qu'il y ait assez de place libre dans le tampon programme PB, c'est-à-dire qu'il y ait un double-mot libre, étant donné qu'une requête automatique travaille sur un double-mot. Bien entendu, le compteur mémoire AMC est augmenté alors de 8 (8 octets du double-mot).

Une telle requête peut être également demandée par la microfonction SIF. Cette microfonction a pour fonction de demander une opération de lecture de 8 octets s'il y a assez de place dans le tampon programme PB et si le contenu du compteur mémoire AMC est vide. Ce compteur mémoire AMC est vide lorsque tous les octets d'une instruction ont été transmis dans le tampon programme PB, c'est-à-dire qu'il n'est plus nécessaire d'émettre une requête de lecture pour aller chercher les octets manquants d'une instruction. Dans

ces conditions, la microfonction SIF permet d'émettre une requête pour alimenter l'instruction qui suit celle dont le traitement est en cours ou terminé.

Le traitement de ces requêtes sera détaillé plus loin dans le cas des transferts d'informations relatives à des données. Il sera également étudié plus loin les conditions d'exécution des différentes microinstructions précitées.

Au cours de l'exécution du programme, des transferts relatifs à des informations sur des données peuvent être demandés. Ce sont alors les tampons de lecture et d'écriture de l'unité DIM qui sont concernés.

Comme pour le tampon programme PB, les tampons de lecture RB0, RB1, RT et d'écriture WB, WT sont associés à un compteur mémoire AMC et à des pointeurs MP, AMP, et BP. Ces tampons sont associés à un second compteur mémoire BC qui contient le nombre d'octets (moins un) restant à échanger entre ces tampons et le bus de liaison BUSA.

Pour expliciter ces transferts, il va être décrit ci-après un exemple d'application en se référant aux figures 5 à 8.

Au cours du déroulement du programme, une instruction peut concerner le transfert de données. Le microprogramme chargé d'exécuter cette instruction va par conséquent être amené à demander, par exemple, la lecture d'une chaîne de 5 octets A, B, C, D et E. Dans un premier exemple, on va considérer que cette chaîne d'octets est enregistrée dans le cache C du processeur P avec cadrage sur un double-mot (figure 5).

Avant d'effectuer un tel transfert entre le cache C et le bus de liaison BUSA, avec passage dans les tampons de lecture RB0, RB1 de l'unité DIM, il est nécessaire de connaître deux informations supplémentaires, à savoir :

- le mode de progression d'adresse (+ ou -), c'est-à-dire savoir si les octets successifs vont être lus avec incrémentation ou décrémentation de l'adresse, et

- l'alignement de la chaîne d'octets pour savoir si elle est justifiée à gauche ou à droite.

Dans l'exemple illustré sur la figure 5, la chaîne de 5 octets à lire est lue en mode de progression d'adresse (+), est justifiée à gauche et est cadrée sur la limite d'un double-mot.

L'initialisation de la lecture de cette chaîne d'octets s'effectue par une microfonction IRD présente dans le microprogramme chargé d'exécuter l'instruction décodée par l'unité PIM.

Lorsque la microfonction IRD est initialisée, des requêtes sont émises par l'unité DIM et l'unité ACM en direction du cache C qui va traiter ces requêtes.

En se reportant à la figure 6, il a été représenté plusieurs cycles de traitement élémentaires i0 à i3, en indiquant dans ces différents cycles l'exécution d'une microfonction après initialisation, le transfert des octets sur le bus de liaison BUSA et dans le tampon de lecture RBO, l'émission des requêtes et l'évolution des différents paramètres associés au tampon de lecture RBO, à savoir BP, BC, MP, AMP et AMC.

Supposons l'initialisation de la microfonction IRD au cycle i0. La prise en compte de cette microinstruction d'initialisation de lecture donne lieu, au cycle i1, à une requête DIRQ adressée par l'unité DIM à l'unité ACM. Au cours de ce même cycle, l'unité ACM émet une requête ACRQ au

cache C qui, dans des conditions optimales, répond dans ce même cycle par un signal RQA en direction de l'unité DIM, signifiant ainsi que la requête ACRQ est acceptée.

L'exécution de la microfonction IRD a pour conséquence de mettre à jour automatiquement les différents paramètres associés au tampon de lecture RB0. Pour l'exemple représenté sur la figure 5, le pointeur BP prend la valeur 0 pour indiquer que le prochain octet à échanger avec le bus de liaison BUSA est l'octet représentatif de la donnée A située à l'octet 0 du double-mot adressé dans le cache C du processeur P. Le compteur BC prend la valeur 4 pour indiquer le nombre d'octets (moins un) restant à échanger avec le bus de liaison BUSA (dans l'exemple il y a 5 octets à transférer). Le pointeur MP prend la valeur 0 pour indiquer que le prochain octet à échanger avec le cache C est le premier octet ou octet 0 du double-mot adressé dans le cache C. Le pointeur AMP prend la valeur 0 pour pointer sur le premier mot de la prochaine requête à échanger avec le cache C, ce premier mot étant le mot 0 du double-mot adressé dans le cache. Le compteur AMC prend la valeur 4 pour indiquer le nombre d'octets (moins un) à échanger avec le cache C lorsque la requête courante sera terminée. Etant donné que cette requête n'a pas encore été exécutée, il y a donc 5 octets à transférer.

Au cycle i2 suivant, le cache C peut signifier par l'émission d'un signal DTA en direction de l'unité DIM, que les octets à transférer sont disponibles sur le bus de liaison BP-DT reliant la cache C à l'unité DIM. Parallèlement, la microfonction du microprogramme en cours qui suit la microfonction IRD est une microfonction GET qui assure le contrôle du transfert des octets en direction du bus de liaison BUSA. Bien évidemment, cette microfonction ne peut être exécutée que si les octets à transférer sont disponibles dans le tampon de lecture RB0 ou sur le bus BP-

DT. Dans le cas présent, le cache ayant émis le signal DTA, cela signifie que la chaîne d'octets à lire est prête à être transférée sur la bus BP-DT. Ainsi, dans le cas où la microfonction GET demande le transfert de 4 octets (4 octets maximum), ceux-ci peuvent être directement envoyés au bus de liaison BP-DT vers le bus de liaison BUSA directement, c'est-à-dire en passant par le multiplexeur M3, le circuit d'alignement des octets CA1 (qui fonctionne comme le circuit d'alignement CA2 décrit précédemment pour le tampon programme), et le multiplexeur M4. Par contre, les 4 octets A,B,C,D ne seront disponibles dans le tampon de lecture RB0 qu'au cycle suivant, c'est-à-dire au cycle i3.

Au cours du cycle i2, seuls le pointeur AMP et le compteur AMC sont mis à jour. Le pointeur AMP prend la valeur 8 pour pointer sur le premier mot de la prochaine requête à échanger avec le cache C, c'est-à-dire l'octet 8 ou le premier octet du double-mot suivant du cache C. Le compteur AMC prend la valeur (- 1). En effet, le nombre d'octets restant à échanger avec le cache C est en fait 0, puisque la requête de lecture concerne un double-mot du cache, double-mot qui contient la totalité de la chaîne d'octets à lire.

Au cycle i3 suivant, les 4 octets A, B, C, D de la chaîne à transférer sont reçus dans la tampon de lecture RB0. Cependant, il faut encore transférer sur le bus de liaison BUSA l'octet E de la chaîne initiale. Ce transfert s'effectue par une seconde microfonction GET. Comme dans le cycle i2, l'octet E et les trois octets suivants sont directement envoyés du bus de liaison BP-DT vers le bus de liaison BUSA, alors que ces 4 mêmes octets seront présents dans le tampon de lecture RB0 au cycle i4 suivant. Au cycle i3, le pointeur BP, le compteur BC et le pointeur MP du tampon de lecture RB0 sont mis à jour. Le pointeur BP prend la valeur 4 pour indiquer que le prochain octet à échanger avec le bus de liaison BUSA est l'octet 4 du double-mot

adressé du cache C. Le compteur BC prend la valeur 0 pour indiquer qu'il reste encore un octet à transférer sur le bus de liaison BUSA (ce compteur prend en fait la valeur 0 puisqu'il indique le nombre d'octets restants moins un). Le pointeur MP prend la valeur 4 pour indiquer le prochain mot à échanger avec le cache C, mot qui commence à l'octet 4 du double-mot adressé du cache C.

Le pointeur BP, le compteur BC et le pointeur MP sont également remis à jour au début du cycle i4. Le pointeur BP prend la valeur 5 pour indiquer que le prochain octet à échanger avec le bus de liaison BUSA est l'octet 5 du double-mot adressé du cache C. Le compteur BC prend la valeur (- 1), étant donné qu'il n'y a plus d'octet à transférer sur le bus de liaison BUSA pour la requête qui a été précédemment émise. Le pointeur MP prend la valeur 8 pour indiquer en fait que le prochain mot à échanger avec le cache C commence à l'octet 8, c'est-à-dire au double mot suivant le double-mot précédemment adressé du cache C.

En se reportant aux figures 7 et 8, il est décrit également le transfert d'une chaîne de 5 octets, mais avec présence d'un décadrage tel, que la chaîne d'octets s'étend de part et d'autre d'une limite de double-mot dans le cache C. Cet exemple diffère du précédent essentiellement par le fait qu'il est nécessaire de demander deux requêtes de lecture au niveau du cache C pour pouvoir transmettre la chaîne d'octets A, B, C, D, E. Après initialisation de la lecture par une microfonction IRD au cycle i0, des requêtes de lecture sont émises au cache C (au cycle i1) comme précédemment, avec mise à jour des paramètres BP, BC, MP, AMP et AMC. Au cycle i2 où l'on demande le transfert des quatre premiers octets A, B, C et D en direction du bus de liaison BUSA sous la commande de la microfonction GET, il est à noter que cette microfonction ne peut pas être exécutée, étant donné que ces 4 octets ne sont pas encore

disponibles.

En effet, au cours de la première requête de lecture, seuls les octets A et B sont concernés puisque présents dans le premier double-mot adressé du cache C. Il faut attendre le cycle i4 pour que la microfonction GET soit effectivement exécutée, les octets C et D de la chaîne étant envoyés directement depuis le bus de liaison BP-DT vers le bus de liaison BUSA, alors que ces deux octets ne seront disponibles qu'au cycle suivant dans le tampon de lecture RB0. Il faudra une deuxième microfonction GET pour assurer le transfert de l'octet E de la chaîne sur le bus de liaison BUSA. Comme précédemment, le transfert des octets de la chaîne passe par le circuit d'alignement CA1 avant que ces octets soient effectivement envoyés sur le bus de liaison BUSA.

Bien entendu, c'est le même microprogramme qui assure le transfert aussi bien de la chaîne d'octets de la figure 5 que de la chaîne d'octets de la figure 7, puisque le microprogrammeur n'a pas à prendre en compte le décadrage éventuel des octets.

Il va être décrit maintenant en référence aux figures 9 et 10, un exemple de transfert d'une chaîne d'octets entre le bus de liaison BUSA et l'ensemble mémoire du système, en considérant cette fois-ci le tampon d'écriture WB de l'unité DIM.

Soit une chaîne de huit octets A, B, C, D, E, F, G, H à transmettre d'une unité de traitement du sous-système CSS vers l'ensemble mémoire du système. Ce transfert va s'effectuer par l'intermédiaire du bus de liaison BUSA reliant les différentes unités du sous-système CSS, et le tampon d'écriture WB de l'unité DIM.

Supposons que cette chaîne d'octets soit décadrée par rapport à une limite de double-mot, comme représenté sur la figure 9. Cette représentation de la chaîne d'octets correspond à celle que prendra cette même chaîne d'octets dans l'ensemble mémoire. Bien entendu, ce décadrage n'est pas pris en compte au niveau du bus de liaison BUSA qui considère cette chaîne de huit octets comme deux mots à transférer à raison de un mot à la fois vers le tampon d'écriture WB de l'unité DIM.

L'initialisation d'un tel transfert s'effectue par une microfonction IWT contenue dans le microprogramme en cours d'exécution et qui demande ce transfert.

Lorsque la microfonction IWT est initialisée, différentes requêtes sont émises par l'unité DIM et l'unité ACM en direction de l'ensemble mémoire du système, notamment au cache C du processeur P chargé de traiter ces requêtes.

En se reportant à la figure 10, il a été représenté plusieurs cycles successifs de traitement élémentaires i0 à i7, en indiquant dans ces différents cycles : l'apparition d'une microfonction après initialisation, l'émission et l'acceptation de requêtes pour l'exécution de cette microfonction, le transfert des octets depuis le bus de liaison BUSA vers le tampon d'écriture WB de l'unité DIM, et depuis ce tampon d'écriture vers l'ensemble mémoire du système, et enfin l'évolution des différents paramètres associés à la gestion du tampon d'écriture WB à savoir : BP, BC, MP, AMP et AMC.

La microfonction IWT est associée à un paramètre, à savoir la longueur de la chaîne d'octets à écrire, soit dans le cas présent huit octets.

Supposons la microfonction IWT initialisée au cycle i0. A ce

cyle i0, les quatre premiers octets A, B, C, D sont disponibles sur le bus de liaison BUSA, et sont disponibles dans le tampon d'écriture WB au cycle il. Il est important de noter que les premiers octets A, B, C, D sont transférés dans le tampon d'écriture WB en passant à travers le circuit d'alignement CA3 (figure 2), afin que ces octets soient écrits dans l'ensemble mémoire en fonction du désalignement possible donné par les premiers bits de poids faible de l'adresse dans l'ensemble mémoire de l'information correspondant à la chaîne d'octets à écrire. Ce circuit d'alignement CA3 fonctionne sur le même principe que le circuit d'alignement CAl précédemment décrit, sauf que l'on considère cette fois-ci, en fonction du décalage par rapport à une limite de mot (dans le cas présent deux ou dix en binaire), les colonnes 0, 1, 2, 3 du tampon d'écriture WB où il faut écrire successivement les quatre octets A, B, C, D.

Comme seuls les quatre premiers octets ont été pris en compte par la microfonction d'initialisation IWT, une microfonction PUT va au moins suivre la microfonction IWT pour assurer le transfert, depuis le bus de liaison BUSA, des autres octets de la chaîne, à raison de quatre octets par microfonction PUT. Bien entendu, une microfonction PUT peut être mise en attente s'il n'y a pas assez de place disponible dans le tampon d'écriture WB.

Dès le cycle il, le tampon d'écriture WB peut faire émettre par l'unité DIM une requête de transfert en direction de l'ensemble mémoire du système, puisque son premier double-mot est plein bien que contenant que les deux premiers octets A, B de la chaîne d'octets à transférer. Les deux octets A, B occupent en effet les deux octets 6, 7 du premier double-mot du tampon d'écriture WB du fait du décadrage indiqué sur la figure 9.

Cette requête DIRQ émise au cycle il par l'unité DIM est

adressée à l'unité ACM qui, au cours de ce même cycle il, émet à l'ensemble mémoire, notamment au cache C, une requête ACRQ. Cette requête, dans des conditions optimales, peut être acceptée par le cache C au cours de ce même cycle il, et traduite par l'émission d'un signal RQA à l'unité DIM signifiant l'acceptation de la requête.

L'exécution de la microfonction IWT a pour conséquence de mettre à jour automatiquement les différents paramètres associés au tampon d'écriture WB.

Dans l'exemple représenté, le pointeur BP prend la valeur A pour indiquer que le prochain octet à échanger avec le bus de liaison BUSA est l'octet représentatif de la donnée E qui devra se situer à l'octet A du tampon d'écriture WB comprenant 16 octets numérotés de 0 à F. Le compteur BC prend la valeur 3 pour indiquer le nombre d'octets (moins un) restant à échanger avec le bus de liaison BUSA (dans l'exemple il reste les quatre octets E, F, G, H, à transférer). Le pointeur MP pointe sur l'octet 6 pour indiquer que le prochain octet à échanger avec le cache C est l'octet 6 du tampon d'écriture WB. Le pointeur AMP prend également la valeur 6 pour pointer sur l'octet du tampon d'écriture WB qui sera à échanger avec le cache C lors de la prochaine requête de transfert entre le tampon d'écriture WB et le cache C. Le compteur AMC prend la valeur 7 pour indiquer le nombre d'octets (moins un) à échanger avec le cache C lorsque la requête courante de transfert sera terminée. Etant donné que cette requête n'a pas encore été exécutée, il y a donc sept octets à transférer.

Au cycle i2 suivant, le cache C peut signifier par l'émission d'un signal DTA en direction de l'unité DIM, qu'il est prêt à recevoir des octets par l'intermédiaire du bus de liaison BP-DT alors disponible et reliant le cache C à l'unité DIM. Dans ces conditions, le second mot du tampon

d'écriture WB qui contient les deux premiers octets A, B de la chaîne, est transmis sur le bus de liaison BP-DT.

Au cours de ce cycle i2, il y a une mise à jour des pointeurs BP et AMP, et les compteurs BC et AMC. Le pointeur BP pointe sur l'octet E du tampon d'écriture WB pour indiquer le prochain octet libre. Le compteur BC prend la valeur (-1), étant donné qu'il n'y a plus d'octets à transférer du bus de liaison BUSA vers le tampon d'écriture WB. En effet, les quatre octets restants E, F, G, et H sont disponibles dès le cycle i2 dans le tampon d'écriture WB par suite de l'exécution de la microfonction PUT au cycle il. Le pointeur AMP pointe sur l'octet 8 du tampon d'écriture WB pour indiquer le prochain octet à transférer au cache C. Cet octet 8 correspond à l'octet C de la chaîne d'octets à transférer.

Le compteur AMC prend la valeur 5 pour indiquer qu'il reste en fait six octets de la chaîne à transférer du tampon d'écriture WB au cache C.

Au cycle i2, l'unité DIM émet automatiquement une requête DIRQ, étant donné que son tampon d'écriture WB a son deuxième double-mot rempli. Comme précédemment, il va y avoir émission d'une requête ACRQ et réception par l'unité DIM d'un signal RQA émis par le cache C et signifiant à l'unité DIM que la requête a été acceptée. A ce cycle i3, le pointeur MP est mis à jour pour pointer sur l'octet 8 du tampon d'écriture WB, indiquant ainsi que c'est le prochain octet à transmettre au cache C.

Au cycle i4, le cache C envoie un signal DTA à l'unité DIM pour lui signifier que le bus de liaison BP-DT est disponible pour transmettre des octets au cache C, à raison de quatre octets par cycle, c'est-à-dire que les octets C, D, E et F de la chaîne sont transmis au cache C. Le pointeur

AMP est mis à jour et pointe sur l'octet 10 ou premier octet du double-mot qui suit le dernier octet de la chaîne d'octets. Le compteur AMC prend la valeur (-1) pour indiquer qu'il n'y aura plus d'octets à transférer dans le cache C, une fois la requête de transfert en cours terminée. En effet, comme une requête concerne un double-mot, les octets G et H qui restent à transférer dans le cache C, seront transférés au cycle i5 qui correspond à la prise en compte du deuxième mot de la requête. A ce cycle i5, le pointeur MP est mis à jour et pointe sur l'octet C du tampon d'écriture WB pour indiquer le premier octet à transférer dans le cache C, soit l'octet G des deux derniers octets G et H de la chaîne. Au cycle i6 suivant, le pointeur MP est mis à jour pour pointer sur l'octet E du tampon d'écriture WB, ou prochain octet à transférer dans le cache C. Cet octet E suit l'octet D du tampon d'écriture WB qui contient le dernier octet H de la chaîne transféré dans le cache C au cours du cycle i5.

Sur la figure 10, il est à noter qu'au cycle i2 peut apparaître une microfonction d'initialisation d'écriture IWT. Cette microfonction ne peut pas en fait être exécutée, puisque à ce cycle i2 le tampon d'écriture n'a pas un double-mot disponible pour recevoir des informations provenant du bus de liaison BUSA. Cette microfonction est donc mise en attente pendant les cycles suivants i3 à i5. Elle ne peut être exécutée qu'à partir du cycle i6, dès l'instant où le transfert de la chaîne d'octets A, B, C, D, E, F, G et H est terminé. Cet exemple montre parfaitement une des synchronisations automatiques effectuées par le système conforme à l'invention.

Une suite de remarques sont à faire pour compléter d'une façon plus générale les exemples précédemment décrits.

- Le désalignement des informations peut être quelconque.

Cependant, dans le cas du tampon programme PB, l'information est en fait alignée sur une frontière de mot ou de demi-mot.

- Les états des différents tampons de l'unité DIM peuvent donner lieu simultanément à des requêtes automatiques de transferts d'informations. Pour régler ce problème, il est prévu des circuits de priorité qui valident, à un moment donné, la requête ayant la plus forte priorité. Avant l'acceptation d'une requête automatique, son émission est conditionnée par l'état des pointeurs AMP et BP associés au tampon demandeur, c'est-à-dire du pointeur pointant sur le premier mot de la prochaine requête à échanger avec l'ensemble mémoire, et le pointeur pointant sur le prochain octet à échanger avec le bus de liaison BUSA.

- Le microprogrammeur, dans le cas du transfert d'une chaîne, peut ne pas connaître la longueur de cette chaîne. Il lui suffit de mettre une boucle de transfert dans son microprogramme, le dernier octet de la chaîne étant automatiquement détecté par l'unité DIM en fonction de l'état du compteur BC associé au tampon concerné (la valeur prise alors par ce compteur est -1).

- Les requêtes de transfert automatiques et non automatiques, ces dernières étant initialisées par des microfonctions présentes dans le microprogramme, sont mises automatiquement en attente si les informations demandées ne sont pas disponibles dans les tampons de l'unité DIM.

- En ce qui concerne les requêtes de transfert automatiques émises par l'unité DIM en fonction de l'état de ses tampons, il est à noter qu'une requête automatique peut être émise au cours du même cycle que celui où s'exécute une microfonction de transfert entre l'ensemble mémoire et un tampon de lecture de l'unité DIM (microfonction GET) ou que celui où s'exécute une microfonction de transfert entre le bus de

liaison BUSA reliant les unités de traitement et l'ensemble mémoire (microfonction PUT), afin d'accélérer la procédure des transferts.

- Il est également à noter que la microfonction SIN associée au tampon programme PB permet d'assurer également le remplissage continuel de ce tampon en émettant des requêtes vers le cache, sous réserve, bien entendu, qu'il n'y ait pas de requête en attente et que le tampon programme PB ait assez de place disponible.

REVENDICATIONS

1. Procédé pour gérer des transferts d'informations entre un ensemble mémoire et les différentes unités de traitement d'un système de traitement numérique de l'information, au cours de l'exécution d'une tâche par le système comprenant au moins un processeur (P) associé à un cache (C), une unité de mémoire principale (MMU), le cache (C) et ladite unité de mémoire principale (MMU) formant ledit ensemble mémoire, une unité d'entrée-sortie (IOC), et un sous-système (CSS) incluant lesdites unités de traitement (ACM, ALM, PIM, CSU, SCM) reliées par au moins un bus de liaison (BUSA), caractérisé en ce qu'il consiste :

- à interposer à l'intérieur du sous-système (CSS), entre l'ensemble mémoire (MMU, C) et les unités de traitement (ACM, ALM, PIM, CSU, SCM) du système, une unité d'interface (DIM) comprenant un ensemble de mémoires tampon dits tampons (PB;RBO,RBl;WB ;RT;WT) par lesquels transitent aussi bien des informations relatives à des instructions d'un programme à exécuter et à transférer vers les unités de traitement (ACM, ALM, PIM, CSU, SCM), que des informations relatives à des données demandées à l'ensemble mémoire (MMU, C) par les unités de traitement (ACM, ALM, PIM, CSU, SCM) ou à écrire dans l'ensemble mémoire (MMU, C),

- à effectuer des transferts automatiques d'informations relatives à des instructions entre l'ensemble mémoire (MMU, C) et ladite unité d'interface (DIM), en avance par rapport à l'exécution proprement dite de ces instructions,

- à effectuer des transferts automatiques de données depuis l'ensemble mémoire (MMU, C) vers ladite unité d'interface (DIM) en avance par rapport à leur utilisation par les unités de traitement (ACM, ALM, PIM, CSU, SCM), et

- à effectuer des transferts automatiques de données depuis

les unités de traitement (ACM, ALM, PIM, CSU, SCM) vers l'ensemble mémoire (MMU, C),

ces transferts automatiques étant indépendants des microprogrammes chargés d'exécuter les instructions relatives au traitement d'une tâche.

2. Procédé selon la revendication 1, caractérisé en ce qu'il consiste à donner au microprogrammeur chargé d'écrire les microprogrammes d'exécution des instructions d'un programme, une image de l'ensemble mémoire (MMU, C) alignée à l'octet et à longueur variable, et à gérer automatiquement, par l'intermédiaire de microfonctions (SIN, IRP, EPR, SIF), tous les problèmes relatifs notamment au cadrage et à la longueur des informations à transférer.

3. Procédé selon la revendication 1 ou 2, caractérisé en ce qu'il consiste à gérer le contenu des tampons (PB ; RBO, RBl ; WB ; RT ; WT) de ladite unité d'interface (DIM) par l'intermédiaire de compteurs (AMC, BC) et de pointeurs (MP, BP,AMP)en fonction de la disponibilité pour ces tampons (PB ; RBO, RBl ; WB ; RT ; WT) à recevoir et à transmettre des informations entre l'ensemble mémoire (MMU, C) et les unités de traitement (ACM, ALM, PIM, CSU, SCM) du système.

4. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'il consiste à faire émettre automatiquement par ladite unité d'interface (DIM), des requêtes de transfert automatique (DIRQ) d'informations relatives à des instructions ou à des données depuis l'ensemble mémoire (MMU, C) vers au moins un tampon de ladite unité d'interface (DIM), dès que ce tampon est vide ou a suffisamment de place disponible pour recevoir au moins une partie de ces informations en minimisant le nombre de requêtes mémoire.

5. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'il consiste à faire émettre

automatiquement par ladite unité d'interface (DIM), des requêtes de transfert automatique (DIRQ) d'informations relatives à des données entre au moins un tampon de ladite unité d'interface (DIM) et l'ensemble mémoire (MMU, C), dès que ce tampon contient au moins déjà une partie de ces informations.

6. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'il consiste à mettre automatiquement en attente l'exécution d'un transfert demandé par un microprogramme, si les informations demandées par ce microprogramme ne sont pas encore disponibles dans le tampon associé de ladite unité d'interface (DIM).

7. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'il consiste à transférer automatiquement des chaînes d'octets entre l'ensemble mémoire (MMU, C) et les unités de traitement (ACM, ALM, PIM, CSU, SCM) sans que le microprogrammeur connaisse au départ la longueur de ces chaînes d'octets, la fin d'une chaîne étant automatiquement détectée au niveau des tampons (PB ; RB0, RB1 ; WB ; RT ; WT) associés de ladite unité d'interface (DIM) par lesquels transitent ces chaînes d'octets.

8. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'il consiste, lorsque plusieurs tampons (PB ; RB0, RB1 ; WB ; RT ; WT) de ladite unité d'interface (DIM) sont émetteurs d'une requête de transfert (DIRQ) d'informations, à gérer la priorité de ces requêtes par des circuits de priorité.

9. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'il consiste à associer un circuit d'alignement (CA3, CA1, CA2) des informations à chaque tampon (WB ; RB0, RB1 ; PB) de ladite unité d'interface (DIM), chaque circuit d'alignement (CA1, CA2, CA3) contrôlant automatiquement le cadrage des informations donné

par les premiers bits de poids faible des adresses mémoire de ces informations.

10. Dispositif pour la mise en oeuvre du procédé tel que défini selon l'une des revendications précédentes, caractérisé en ce que l'unité d'interface (DIM) précitée comprend un tampon programme (PB) par lequel transitent des informations relatives aux instructions d'un programme à exécuter, au moins un tampon de lecture (RB0, RB1) par lequel transitent des informations relatives à des données transmises depuis l'ensemble mémoire (MMU, C) du système vers les unités de traitement (ACM, ALM, PIM, CSU, SCM) du système, au moins un tampon d'écriture (WB) par lequel transitent des informations relatives à des données transmises depuis les unités de traitement vers l'ensemble mémoire (MMU, C) du système, ces tampons (PB ; RB0, RB1 ; WB) étant reliés d'une part audit ensemble mémoire (MMU, C) et d'autre part à au moins certaines unités de traitement (ACM, ALM, PIM, CSU, SCM) du système.

11. Dispositif selon la revendication 10, caractérisé en ce que les tampons (PB ; RB0, RB1 ; WB) de ladite unité d'interface (DIM) sont gérés par des compteurs (AMC, BC)et par des pointeurs (MP, BP, AMP) mis à jour automatiquement par des microinstructions.

12. Dispositif selon la revendication 11, caractérisé en ce que les tampons (PB ; RB0, RB1 ; WB) de ladite unité d'interface (DIM) sont associés à un pointeur (BP) pointant sur le prochain octet à échanger avec le bus de liaison (BUSA) reliant les unités de traitement (ACM, ALM, PIM, CSU, SCM) du système, un pointeur (MP) pointant sur le prochain octet à échanger avec l'ensemble mémoire (MMU, C) du système, un pointeur (AMP) qui pointe sur le premier mot de la prochaine requête (DIRQ) de transfert à échanger avec l'ensemble mémoire (MMU, C) du système, et un compteur mémoire (AMC) pour compter le nombre d'octets restants à échanger avec l'ensemble mémoire (MMU, C) lorsque la requête de transfert (DIRQ) en cours de traitement sera terminée.

13. Dispositif selon la revendication 11 ou 12, caractérisé en ce que les tampons d'écriture (WB) et de lecture (RB0, RB1) de ladite unité d'interface (DIM) sont associés également à un compteur mémoire (BC) qui contient le nombre d'octets restants à échanger avec le bus de liaison (BUSA) reliant les différentes unités de traitement (ACM, ALM, PIM, CSU, SCM) du système.

14. Dispositif selon l'une des revendications 10 à 13, caractérisé en ce que chaque tampon (RB0, RB1 ; PB ; WB) de ladite unité d'interface (DIM) est associé à un circuit d'alignement (CA1, CA2, CA3) des informations pour tenir compte automatiquement du cadrage de ces informations, cadrage donné par les premiers bits de poids faible de l'adresse mémoire de ces informations.

15. Dispositif selon l'une des revendications 10 à 14, caractérisé en ce que chaque tampon (RB0, RB1 ; PB, WB) de ladite unité d'interface (DIM) a une longueur correspondant au moins à un double-mot, un mot ayant quatre octets.

Fig. 1

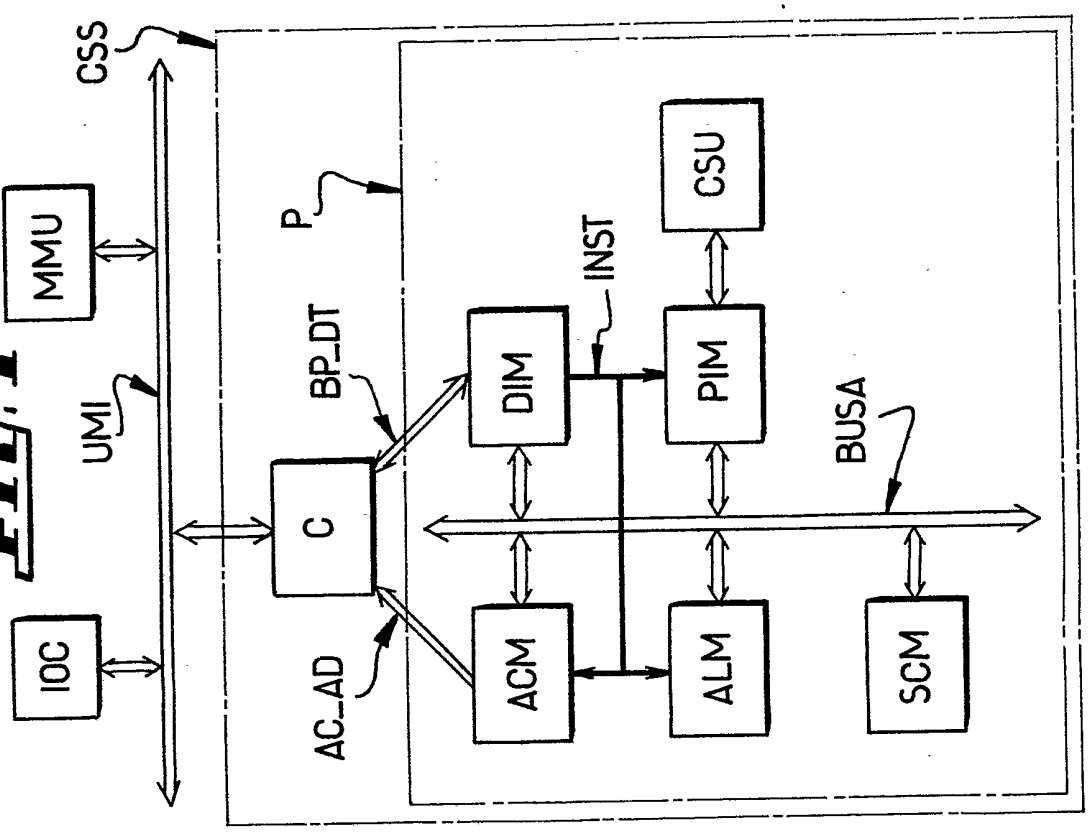


Fig. 3

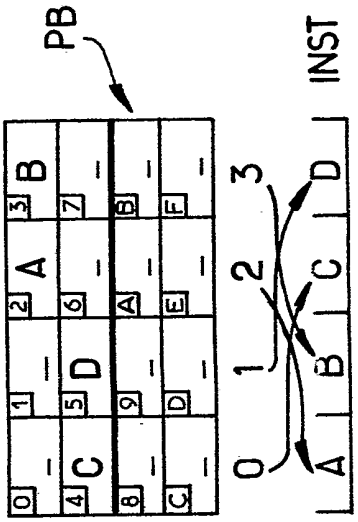


Fig. 4

BP	Octets sur INST			
0 0	0	1	2	3
0 1	1	2	3	0
1 0	2	3	0	1
1 1	3	0	1	2

Fig. 2

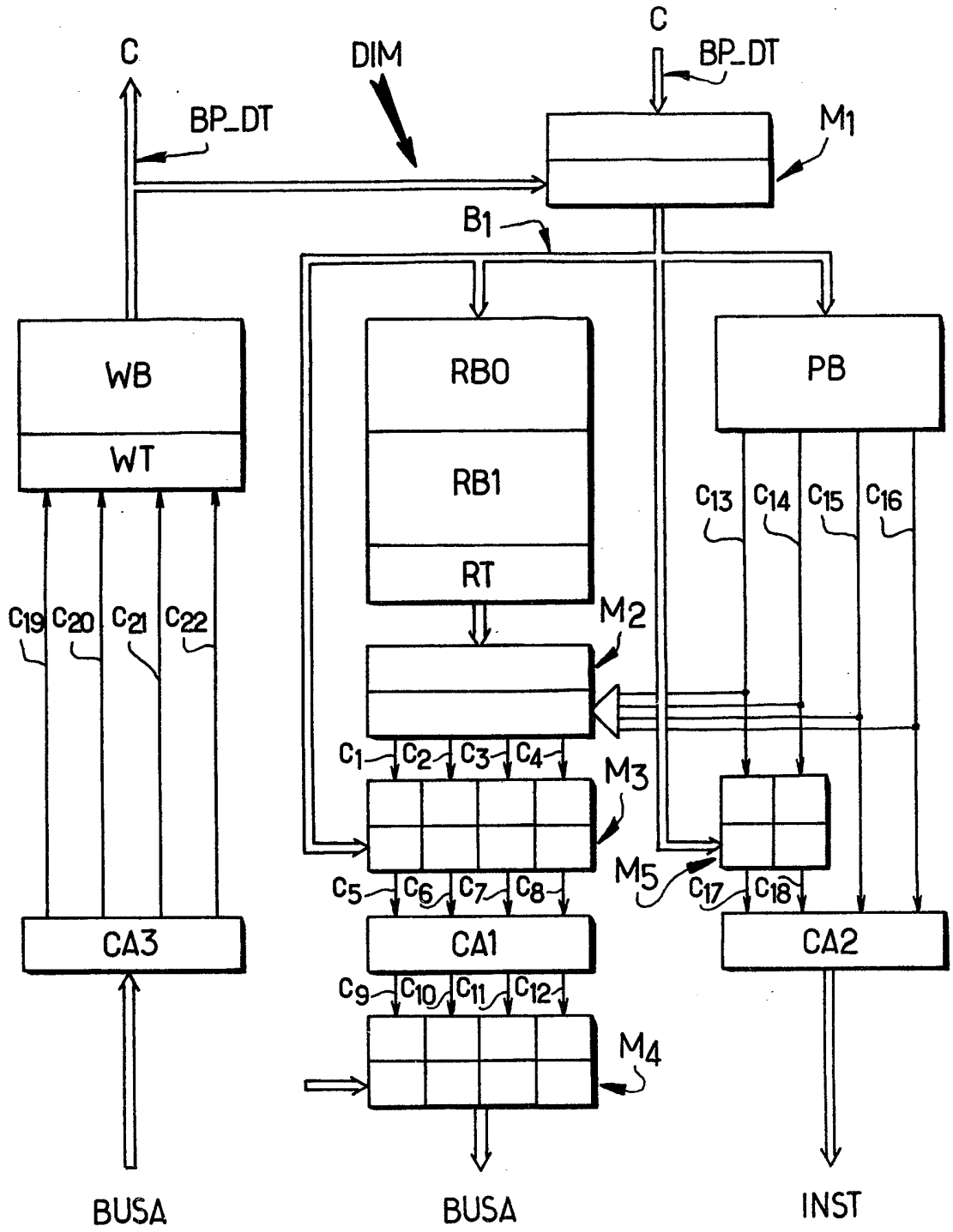
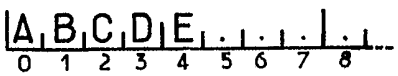

C
BP_DT
DIM
C
BP_DT
M1
B1
WB
WT
RB0
RB1
RT
PB
C19 C20 C21 C22
C13 C14 C15 C16
M2
C1 C2 C3 C4
M3
M5
C5 C6 C7 C8
C17 C18
CA3
CA1
CA2
C9 C10 C11 C12
M4
BUSA
BUSA
INST

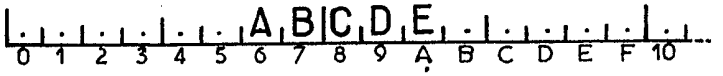
Fig 5



micro-instruction	i_0	i_1	i_2	i_3	i_4
	IRD		GET	GET	
BUSA			ABCD	E . . .	
Requêtes		DIRQ ACRQ RQA	DTA	DTA	
RB0				ABCD	ABCD E . . .
BP		0	–	4	5
BC		4	–	0	-1
MP		0	–	4	8
AMP		0	8	–	–
AMC		4	-1	–	–

Fig 6

Fig 7



micro-instruction	i_0	i_1	i_2	i_3	i_4	i_5	i_6
	IRD		GET			GET	
BUSA					ABCD	E . . .	
Requêtes		DIRQ ACRQ RQA	DIRQ DTA	DIRQ ACRQ RQA	DTA		
RB0				. . AB		. . AB CDE.	
BP		6	–	–	–	A	B
BC		4	–	–	–	0	-1
MP		6	–	8	–	C	–
AMP		6	8	–	10	–	–
AMC		4	2	–	-1	–	–

Fig 8

Fig. 9

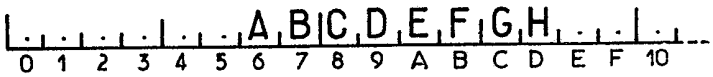


Fig. 10

	i_0	i_1	i_2	i_3	attente i_4	i_5	i_6
micro-instruction	IWT	PUT	IWT	.....	.....	IWT	IWT
BUSA	ABCD	EFGH					
Requêtes		DIRQ ACRQ RQA	DIRQ DTA	DIRQ ACRQ RQA	DTA		
WB		..AB CD..	..AB CDEF GH..				
BP_DT			..AB		CDEF	GH..	
BP		A	E	–	–	–	–
BC		3	-1	–	–	–	–
MP		6	–	8	–	C	E
AMP		6	8	–	10	–	–
AMC		7	5	–	-1	–	–