(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 808 902**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **01 05866**

(51) Int Cl[7] : **G 06 F 12/02,** G 06 F 12/08

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 02.05.01.

(30) **Priorité :** 04.05.00 US 09565017.

(43) **Date de mise à la disposition du public de la demande :** 16.11.01 Bulletin 01/46.

(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) :** *HEWLETT PACKARD COMPANY* — US.

(72) **Inventeur(s) :** LESARTRE GREGG B et JOHNSON DAVID JEROME.

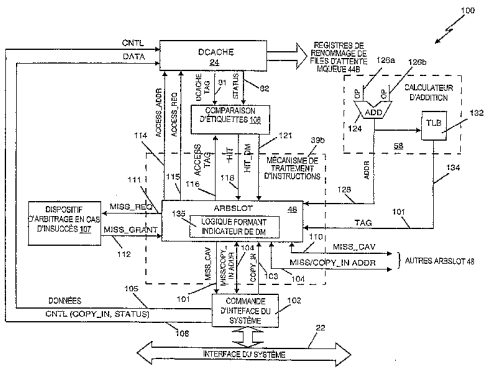
(73) **Titulaire(s) :**

(74) **Mandataire(s) :** REGIMBEAU.

(54) SYSTEME DE PRE-EXTRACTION SPECULATIVE DE DONNEES DANS UN SYSTEME DE PROCESSEUR D'EXECUTION NON ORDONNEE D'INSTRUCTIONS.

(57) Un dispositif (100) pour réduire le temps de latence en cas d'insuccès dans une antémémoire comprend des fentes (48) contenant des adresses de lignes de transmission de données mémorisées dans une antémémoire (24) et un système hiérarchique de mémoire, au moins une fente de pré-extraction configurée pour déterminer une ligne additionnelle de transmission de données devant être pré-extraite dans l'antémémoire lors d'un insuccès dans l'antémémoire pour une première ligne, et une logique associée aux fentes pour indiquer que la ligne additionnelle est déjà demandée par une fente (48) à partir du système hiérarchique de mémoire.

Application notamment à l'optimisation de performances dans un système de processeur.



FR 2 808 902 - A1

La présente invention concerne d'une manière générale des opérations et des architectures de processeurs d'ordinateurs. Plus particulièrement la présente invention concerne l'optimisation de performances au moyen de la pré-extraction et du prévidage spéculatifs de données dans un système de processeur, dans lequel des instructions peuvent être exécutées d'une manière non ordonnée.

Un processeur à hautes performances, par exemple un processeur super-scalaire, dans lequel deux ou un plus grand nombre d'opérations scalaires sont exécutées en parallèle, peut être conçu de manière à exécuter des instructions d'une manière non ordonnée, c'est-à-dire dans un ordre qui est différent de celui qui est défini par le programme exécuté dans le processeur. C'est-à-dire que, dans le système de processeur à hautes performances, les exécutions sont exécutées lorsqu'elles peuvent être exécutées plutôt que lorsqu'elles apparaissent selon la séquence définie par le programme. De façon typique, après l'exécution non ordonnée d'instructions, les résultats sont finalement réarrangés de manière à correspondre à l'ordre correct des instructions, avant le renvoi des résultats au programme qui est exécuté dans le processeur.

Des exemples d'architectures de processeur, qui exécutent des instructions d'une manière non ordonnée sont décrits dans le brevet US N°5 758 178 (délivré le 26 Mai 1998 et ayant pour titre "Miss Tracking System and Method"), le brevet US N°5 761 713 (délivré le 2 Juin 1998 et ayant pour titre "Address Aggregation System and Method for Increasing Throughput to a Multi-Banked Data Cache From a Processor by Concurrently Forwarding an Address to Each Bank"), le brevet US N°5 838 942 (délivré le 17 Novembre 1998 et ayant pour titre "Panic Trap System and Method"), le brevet US N°5 809 275 (déposé le 15 Septembre 1998 et ayant pour titre "Store-to Load Hazard Resolution System and Method for a Processor that Executes Instructions Out

of Order"), le brevet US N°5 799 167 (délivré le 25 Août 1998 et ayant pour titre "Instruction Nullification System and Method for a Processor that Executes Instructions Out of Order"), tous au nom de Gregg Lesartre qui est l'un des présents inventeurs, et qui sont tous incorporés expressément par référence, dans leur totalité.

Comme cela est décrit de façon plus détaillée par exemple dans le brevet US N°5 758 178, un système de processeur d'exécution non ordonnée peut inclure un ou plusieurs processeurs, chacun possédant une file d'attente de mémoire (MQUEUE) pour recevoir et exécuter des instructions qui sont envoyées pour des accès à l'antémémoire (DCACHE) ou au système hiérarchique de mémoire. La file d'attente MQUEUE inclut une pluralité de mécanismes de traitement d'instructions pour recevoir et exécuter des instructions de mémoire respectives de manière non ordonnée. Chaque mécanisme de traitement d'instruction inclut un registre d'instruction pour mémoriser une instruction et une fente de tampon de réarrangement d'adresses (ARBSLOT) servant à mémoriser l'adresser de données des résultats de l'exécution d'instructions. Ce qui est important, une logique d'indication, qui est fonction d'un insuccès (DM), dans chaque ARBSLOT empêche une demande depuis sa fente respective ARBSLOT en direction de ce système hiérarchique de mémoire pour des données d'insuccès qui sont absentes de l'antémémoire DCACHE lorsqu'une autre fente ARBSLOT a déjà demandé les données d'insuccès à partir du système hiérarchique de mémoire.

En particulier, à titre d'exemple, la figure 1 représente un schéma-bloc des parties importantes du système d'ordinateur servant à illustrer le fonctionnement de la partie de mécanisme 39b de traitement d'instructions de la file d'attente MQUEUE. La file d'attente MQUEUE inclut une ou plusieurs fentes ARBSLOT 48 (dont l'une est représentée). Lorsqu'une fente ARBSLOT 48 demande une ligne

d'antémémoire à partir de l'antémémoire DCACHE 24, le signal de positionnement de la fente ARBSLOT 48 ACCESS_REQ 115 accompagné d'une adresse ACCESS_ADDR 114. Dans le cas où il existe un succès potentiel dans l'antémémoire 24, l'indicateur d'état 82 (ou des indicateurs d'états si la mémoire est associative) reflète une ou des lignes valables de l'antémémoire. En outre le mécanisme 108 de comparaison d'étiquettes lit la ou les étiquettes DCACHE_TAG(s) 81 et les compare à l'étiquette ACCESS_TAG 116 associée à l'adresse d'accès ACCESS_ADDR 114. Lorsqu'il existe une correspondance, le mécanisme 108 de comparaison d'étiquettes en conclut qu'il existe un succès et supprime le signal ~HIT 118 pour indiquer un succès, ce qui amène la fente ARBSLOT 48 à se marquer elle-même. Le résultat de l'opération est conservé dans un registre de renommage (non représenté) jusqu'à ce que l'instruction se retire, lorsqu'elle est envoyée à un registre architectural (non représenté).

Lorsque l'accès à l'antémémoire conduit à un insuccès dans l'antémémoire, par exemple sur la base d'un indicateur d'état 82 indiquant une ou des lignes d'antémémoire non valables ou sinon lorsque l'étiquette DCACHE_TAG(s) 81 ne concorde pas avec l'étiquette ACCESS_TAG 116, alors le mécanisme 108 de comparaison d'étiquettes positionne le signal ~HIT 118 pour indiquer un insuccès concernant la fente ARBSLOT 48. En supposant qu'il s'agit de la première fente ARBSLOT 48 pour essayer d'accéder à cette ligne de transmission de données d'insuccès, la logique formant indicateur DM 135 provoque la délivrance du signal de demande conduisant à un insuccès MISS_REQUEST 111 au dispositif d'arbitrage d'insuccès 107. Le dispositif d'arbitrage d'insuccès 107 réalise un arbitrage en donnant la priorité à différentes demandes d'insuccès qui peuvent être produites par les différentes fentes ARBSLOT 48. Eventuellement le dispositif d'arbitrage

d'insuccès 107 délivre un signal MISS_GRANTED 112 pour accorder la demande conduisant à un insuccès. Ce signal est envoyé à la fente ARBSLOT 48 qui à son tour positionne le signal de commande d'insuccès MISS_CAV 101 en l'envoyant à la commande 102 d'interface du système. A son tour la commande 102 d'interface de système présente une demande de mémoire au système hiérarchique de mémoire (non représenté) pour la ligne de données sur la base de l'adresse MISS/COPY_IN ADDR 104 qui est transmise de la fente ARBSLOT 48 à la commande 102 d'interface du système.

Une fois que la ligne de transmission de données est transférée depuis le système hiérarchique de mémoire à la commande 102 d'interface du système, la commande 102 d'interface du système transfère la ligne de transmission de données à l'antémémoire DCACHE 24, comme indiqué par la flèche de référence 105, positionne le signal COPY_IN dans l'antémémoire DCACHE 24, et délivre les bits d'état à la mémoire DCACHE 24. Simultanément, la commande 102 d'interface du système positionne le signal de commande COPY_IN 103 sur les fentes ARBSLOT 48 et positionne l'adresse associée dans MISS/COPY_IN ADDR 104 sur les fentes ARBSLOT 48.

Si une autre fente ARBSLOT 148 essaie d'accéder à l'antémémoire DCACHE 24 pour une ligne de transmission de données d'insuccès, qui est actuellement demandée à partir du système hiérarchique de mémoire, alors la fente ARBSLOT particulière 48 est avisée par l'indicateur d'état 82, étant donné que l'indicateur d'état 82 indique un état d'insuccès en cours ou bien que la ligne de l'antémémoire est demandée par une autre fente ARBSLOT 48. Par conséquent une demande de mémoire redondante pour une ligne de transmission de données, qui a déjà été demandée, est évitée. On peut trouver une description plus détaillée de la file d'attente de mémoire (MQUEUE) et l'indicateur DM 135 dans les brevets US mentionnés précédemment, par exemple le

brevet US 5 758 178.

Bien que les processeurs modernes à hautes performances, par exemple le processeur super-scalaire décrit précédemment, possèdent une durée d'exécution d'instructions nettement améliorée, un long temps d'accès en mémoire constitue encore une gêne importante pour un processeur qui fonctionne à sa vitesse maximale. Si des demandes de données peuvent être satisfaites à partir de l'antémémoire, des retards associés à un accès au système hiérarchique de mémoire plus lent -- désignés habituellement sous l'expression temps de latence en cas d'insuccès dans l'antémémoire -- peuvent être évités. Par conséquent la réduction du nombre d'insuccès dans l'antémémoire est un objectif dans des types de processeurs à hautes performances.

En outre, dans des systèmes multiprocesseurs, lorsqu'un processeur demande une ligne de transmission de données, un contrôle de cohérence est requis pour déterminer si les antémémoires respectives des autres processeurs contiennent la ligne de données demandée et/ou si une réécriture (ou un vidage) de la ligne de transmission de données en direction du système hiérarchique de mémoire est requise, c'est-à-dire si une réécriture (ou un balayage) de la ligne de données dans un système hiérarchique de mémoire est requis, par exemple lorsque la ligne de transmission de données a été modifiée par le processeur particulier qui possède cette ligne de transmission de données. Le contrôle de cohérence ajoute des retards à des accès en mémoire - désignés ci-après comme étant le temps d'attente de contrôle de cohérence.

Une pré-extraction et un prévidage spéculatifs sont basés sur une théorie de localité bien connue, désignée comme étant la théorie de localité spatiale, qui observe que, lorsque le processeur accède à des informations, l'information, dont les adresses sont proches de

l'information d'accès, tend à faire également l'objet d'un accès. Ceci est particulièrement vrai lorsque l'opération de chargement ou de mémorisation, qui a provoqué l'insuccès en antémémoire est une partie d'une séquence de code d'instructions, qui accède à une longueur d'enregistrement supérieure à celle d'une ligne de l'antémémoire, ou bien lorsque la séquence de code d'instructions se réfère à des données qui englobent une multiplicité de lignes de transmission de données. Dans un système utilisant la pré-extraction et/ou le prévidage (plutôt que l'extraction et/ou le vidage) seules des données, (auquel l'accès est actuellement réalisé) dans (ou à partir) de l'antémémoire, un bloc de données (ou une ou plusieurs lignes de l'antémémoire) situé au voisinage, incluant des données auxquelles un accès est actuellement réalisé, peuvent être introduites dans l'antémémoire (et/ou être vidées de l'antémémoire). Cette pré-extraction et ce prévidage spéculatifs de lignes supplémentaires de transmission de données dans (ou à partir) de l'antémémoire avant qu'ils soient demandés par des instructions ultérieures de référence en mémoire, peuvent masquer au moins un certain temps de latence en cas d'insuccès dans l'antémémoire et le temps de latence de contrôle de cohérence est par conséquent amélioré de la performance globale du système de processeur.

Mais malheureusement jusqu'à présent il n'existait aucune solution connue pour réaliser dans des lignes de transmission de données à pré-extraction et/ou prévidage de lignes de transmission de données dans des processeurs, qui exécutent l'exécution d'instructions d'une manière désordonnée. Dans un système utilisant une pré-extraction ou un prévidage spéculatif décrit précédemment, chaque demande de mémoire additionnelle résultant d'une exécution non ordonnée d'instructions implique une transaction de mémoire qui requiert un transfert d'un

certain nombre de lignes de transmission de données (plutôt qu'une seule ligne de transmission de données sans la pré-extraction ou le prévidage d'une ou de lignes supplémentaires de transmission de données) et peut conduire à un trafic même accru de façon supplémentaire dans le bus du système, peut modifier l'utilisation excessive de la largeur de bande de l'interface du système et peut par conséquent compromettre la performance du système.

C'est pourquoi, il est nécessaire de disposer d'un système et d'un procédé efficaces de pré-extraction d'une ou plusieurs lignes de transmission de données à partir d'un système hiérarchique de mémoire en direction d'une antémémoire sans compromettre la performance d'un système de traitement non ordonné.

Il est également nécessaire de disposer d'un système et d'un procédé efficaces de pré-extraction de ligne de données d'une hiérarchie de mémoires vers une mémoire cache tout en minimisant les requêtes mémoire multiples redondantes dans le cas d'insuccès dans l'antémémoire dans un système de traitement non ordonné.

Il est également nécessaire de disposer d'un système et d'un procédé efficaces de prévidage d'une ou plusieurs lignes de transmission de données à partir d'une antémémoire dans un système de processeurs d'exécution désordonnée d'instructions multiples, sans accroître la complexité du système, et de ce fait réduire le temps de latence de cohérence du système.

Conformément aux principes de la présente invention, un dispositif pour réduire le temps de latence d'insuccès dans l'antémémoire dans un système d'exécution non ordonnée d'instructions comprend une pluralité de fentes de tampons de réarrangement d'adresses, dans chacune desquelles est mémorisée une adresse correspondant à une ligne de transmission de données d'une pluralité de lignes

de transmission de données, chacune de ladite pluralité de lignes de transmission de données étant mémorisée dans au moins une antémémoire et un système hiérarchique de mémoire; au moins une fente de pré-extraction configurée pour, lors de la détection d'un insuccès d'une première ligne de transmission de données dans l'antémémoire, résultant d'une demande d'accès provenant d'au moins une fente de ladite pluralité de fentes de tampons de réarrangement d'adresses, déterminer au moins une ligne additionnelle de transmission de données devant être pré-extraite dans ladite antémémoire à partir dudit système hiérarchique de mémoire; et une logique associée à ladite au moins une fente de pré-extraction, ladite logique étant configurée de manière à fournir une indication du fait que ladite au moins une ligne additionnelle de transmission de données est déjà demandée à partir de ladite hiérarchie de mémoire par l'une quelconque de ladite pluralité de fentes de tampons de réarrangement d'adresses.

Selon une autre caractéristique de l'invention, ladite au moins une fente de pré-extraction comprend une logique d'adresses adjacentes configurée de manière à délivrer une ou plusieurs adresses supplémentaires correspondant à ladite au moins une ligne additionnelle de transmission de données, ladite au moins une ligne additionnelle de transmission de données possédant un emplacement de mémoire adjacent à ladite première ligne de transmission de données.

Selon une autre caractéristique de l'invention, ladite logique d'adresses adjacentes reçoit une première adresse correspondant à ladite première ligne de transmission de données et délivre une ou plusieurs adresses additionnelles par inversion d'un ou de plusieurs bits dans ladite première adresse.

Selon une autre caractéristique de l'invention, le dispositif pour réduire le temps de latence en cas

d'insuccès dans une antémémoire comporte en outre : un circuit de verrouillage d'occupation comportant une entrée de positionnement entre l'effacement, ledit circuit de verrouillage d'occupation étant configuré de manière à délivrer un signal d'occupation, et ledit signal d'occupation étant actif lorsque ladite entrée de positionnement est déclenchée, et inactif lorsque ladite entrée d'effacement est déclenchée; et un registre configuré de manière à mémoriser un indice d'antémémoire et une étiquette, ces deux éléments étant dérivés d'une adresse reçue de la part de ladite logique d'adresses adjacentes, ledit registre recevant ladite adresse de la part de ladite logique d'adresses adjacentes lors de la réception d'un signal de mise à jour, ledit signal de mise à jour étant produit par inversion dudit signal d'occupation.

Selon une autre caractéristique de l'invention, le dispositif pour réduire le temps de latence en cas d'insuccès dans une antémémoire comporte en outre : une logique de décodage pour recevoir un signal valable d'adresse d'insuccès dans l'antémémoire et un type de transaction, ladite logique de décodage étant configurée de manière à déclencher ladite entrée de positionnement dudit circuit de verrouillage d'occupation lorsque ledit signal valable d'adresse d'insuccès dans l'antémémoire introduit indique qu'une demande de transaction est faite audit système hiérarchique de mémoire, et ledit type de transaction introduit indique que ladite demande de transaction est effectuée en raison d'un insuccès d'accès dans l'antémémoire.

En outre, conformément à un autre aspect des principes de la présente invention, un procédé de réduction du temps de latence en cas d'insuccès dans une antémémoire dans un système d'exécution non ordonnées d'instructions comprend : la détection de l'apparition d'un insuccès d'une première ligne de données dans l'antémémoire; le calcul

d'une adresse d'au moins une ligne additionnelle de transmission de données devant être préalablement extraites dans une antémémoire à partir d'un système hiérarchique de mémoires; et déterminer si une demande faite précédemment pour ladite au moins une ligne additionnelle de transmission de données provenant de ladite hiérarchie de mémoire est en cours.

Selon une autre caractéristique de l'invention, ladite étape de calcul de ladite adresse de ladite au moins une ligne additionnelle de transmission de données comprend : l'inversion d'un ou de plusieurs bits d'une adresse de ladite première ligne de transmission de données.

Selon une autre caractéristique de l'invention, le procédé pour réduire le temps de latence en cas d'insuccès dans l'antémémoire consiste en outre, dans le cas où ladite demande formulée précédemment est en cours, à empêcher qu'une demande de mémoire pour ladite au moins une ligne additionnelle de transmission de données soit à nouveau présentée.

Selon une autre caractéristique de l'invention, le procédé pour réduire le temps de latence en cas d'insuccès dans l'antémémoire consiste en outre, si ladite demande formulée précédemment n'est pas en cours, à délivrer une demande pour ladite au moins une ligne additionnelle de transmission de données à partir dudit système hiérarchique de mémoire.

Selon une autre caractéristique de l'invention, le procédé pour réduire le temps de latence en cas d'insuccès dans l'antémémoire consiste en outre à : déterminer si ladite au moins une ligne additionnelle de transmission de données est présente dans ladite antémémoire; si ladite au moins une ligne additionnelle de transmission de données est présente dans ladite antémémoire, empêcher qu'une demande de mémoire pour ladite au moins une ligne

additionnelle de transmission de données soit à nouveau formulée; et si ladite au moins une ligne additionnelle de transmission de données n'est pas présente dans ladite antémémoire, délivrer une demande pour ladite au moins une ligne additionnelle de transmission de données à partir dudit système hiérarchique de mémoire.

D'autres caractéristiques et avantages de la présente invention ressortiront de la description donnée ci-après prise en référence aux dessins annexés, sur lesquels :

- la figure 1 est un schéma-bloc représentant les parties importantes d'un système d'ordinateur existant possédant un processeur d'exécution non ordonnée d'instructions;

- la figure 2 est un schéma-bloc d'une forme de réalisation prise à titre d'exemple de la fente de pré-extraction/prévidage (DPRESLOT) conformément aux principes de la présente invention;

- la figure 2A est un schéma-bloc d'une forme de réalisation prise à titre d'exemple de la logique d'arbitrage de port de l'antémémoire conformément à une forme de réalisation préférée de la présente invention;

- la figure 3 est un organigramme d'un mode de mise en oeuvre pris à titre d'exemple du processus de pré-extraction conformément aux principes de la présente invention;

- la figure 4 est un schéma-bloc d'une forme de réalisation prise à titre d'exemple de la fente de contrôle de cohérence de l'antémémoire (CCCSLOT) conformément aux principes de la présente invention; et

- la figure 5 est un organigramme d'un mode de mis en oeuvre pris à titre d'exemple du processus de prévidage conformément aux principes de la présente invention.

A titre de simplification et à des fins d'illustration, les principes de la présente invention vont être

décrits en se référant principalement à une forme de réalisation prise à titre d'exemple, notamment en référence à un exemple dans lequel un agencement spécifique du circuit est réalisé. Cependant un spécialiste ordinaire de la technique pourra aisément reconnaître que les mêmes principes sont également applicables et peuvent être appliqués à d'autres types de circuits et que n'importe quelle variation de cette sorte se situe dans le cadre de variantes de réalisation qui restent dans le cadre de la présente invention.

Conformément aux principes de la présente invention, une fente de pré-extraction/prévidage (DPRESLOT) est pourvue d'une file d'attente de mémoire (MQUEUE) du processeur d'exécution désordonné. La fente DPRESLOT contrôle les transactions entre une interface du système, par exemple le bus du système, une adresse (ARBSLOT) du tampon d'enregistrement d'adresses et/ou entre l'interface du système et la fente de contrôle de cohérence de l'antémémoire (CCCSLOT). Lorsqu'un insuccès dans l'antémémoire est détecté, la fente DPRESLOT provoque la pré-extraction d'une ou plusieurs lignes de l'antémémoire en plus de la ligne de transmission de données, qui a provoqué l'insuccès actuel dans l'antémémoire, à partir du système hiérarchique de mémoire pour son introduction dans l'antémémoire (DCACHE) d'une manière anticipant les données additionnelles qui seraient demandées dans un futur proche. Lorsqu'une réécriture dans l'antémémoire est détectée en tant que résultat du contrôle de cohérence de l'antémémoire, la fente DPRESLOT provoque le prévidage d'une ou plusieurs lignes de l'antémémoire, en plus de la ligne de transmission de données qui est actuellement réécrite, à partir du système hiérarchique de mémoire à partir de l'antémémoire respective (DCACHE) du processeur qui possède cette ligne, d'une manière anticipant le fait que les données supplémentaires seraient requises par le

processeur demandeur dans le futur proche. Une logique contenue dans la fente DPRESLOT empêche une demande d'accès dans l'antémémoire pour les données additionnelles lorsqu'une autre demande de données a déjà été faite. La pré-extraction et le prévidage spéculatifs des lignes supplémentaires de l'antémémoire réduisent le temps de latence en cas d'insuccès dans l'antémémoire et le temps de latence du contrôle de cohérence d'un processeur d'exécution non ordonnée d'instructions.

En particulier, conformément à un mode de mise en oeuvre préféré de la présente invention, une ou plusieurs fentes DPRESLOT sont ajoutées au mécanisme de traitement d'instructions 39b (figure 1). Sinon une ou plusieurs des fentes ARBSLOT représentées sur la figure 1 peuvent être modifiées pour assumer les fonctions de la fente DPRESLOT, qui vont être maintenant décrites de façon plus détaillée.

La figure 2 représente un schéma-bloc d'une forme de réalisation prise à titre d'exemple de la fente de pré-extraction/prévidage (DPRESLOT) 200 conformément aux principes de la présente invention, qui inclut un registre 136 pour mémoriser un indicateur d'insuccès (~HIT) 136a conditionné par le signal ~HIT 118 provenant du mécanisme de comparaison d'étiquette 108 (figure 1), un indice d'antémémoire 136b et une étiquette d'adresse réelle (TAG) 136c, qui sont reçus respectivement en tant qu'adresse ADDR 128 et en tant qu'étiquette TAG 134, à partir de la logique d'adresses adjacentes 213 et facultativement dans un mode de mise en oeuvre préféré de la présente invention, une mémoire du type d'insuccès 136d pour conserver un drapeau de mémorisation (STORE) reçu de la part d'une entrée 214 du signal MISS_STORE. Le drapeau à un seul bit STORE indique si l'instruction d'accès en mémoire, qui est traitée, exécute une opération de lecture ou une opération d'écriture, et est dérivée de l'instruction actuellement en cours de traitement dans le mécanisme 39b de traitement

d'instructions (figure 1). Le drapeau STORE est utilisé par la mémoire DCACHE 24 pour maintenir le fonctionnement d'antémémoire, en rapport avec la ou les lignes de transmission de données pré-extraites, compatible avec l'instruction d'accès en mémoire qui est exécutée.

La logique d'adresses adjacentes 213 reçoit l'adresse présente dans MISS/COPY_IN ADDR 104, qui fait partie de l'interface transactionnelle entre le mécanisme 39b de traitement d'instructions (figure 1) et la commande 102 d'interface du système (figure 1). La logique d'adresses adjacentes 213 produit des adresses qui sont adjacentes à l'adresse reçue de la part de MISS/COPY_IN ADDR 104, par exemple par inversion d'un ou de plusieurs bits de poids inférieur de l'adresse reçue ou bien moyennant l'utilisation d'un compteur pour produire un certain nombre d'adresses. Dans cette forme de réalisation prise à titre d'exemple, le bit de poids le moins significatif (LSB) de l'adresse reçue est inversé pour produire une seule adresse située en un emplacement directement adjacent à l'adresse reçue, c'est-à-dire précédant ou succédant directement à cette adresse.

La ou les adresses adjacentes ainsi produites sont délivrées dans la ligne ADDR 128 pour être mémorisées dans le CACHE INDEX (l'indice d'antémémoire) 136b du registre 136. La logique d'adresses adjacentes 213 délivre également l'étiquette TAG 134, qui est un numéro de page réel (RPN) associé à l'adresse adjacente dans la forme de réalisation préférée, pour sa mémorisation dans un emplacement TAG (étiquette) 136c du registre 136. Le registre 136 reçoit un signal de mise à jour 212. Lorsque le signal de mise à jour 212 est actif, le registre 136 met à jour son contenu, c'est-à-dire le contenu de chacune des zones, à savoir ~HIT 136a, la zone CACHE INDEX 136b, l'étiquette TAG 136c et la mémoire STORE 136d.

Le signal de mise à jour 212 est délivré par

l'inverseur 219, qui reçoit comme signal d'entrée un signal d'occupation BUSY 204 de la part du circuit de verrouillage d'occupation 203. Le circuit de verrouillage d'occupation 203 peut comporter une bascule bistable de positionnement et de remise à l'état initial (S-R) et comporte deux entrées, une entrée de positionnement SET 205 et une entrée d'effacement CLR 206, qui respectivement positionne et ramène à zéro ou supprime le signal BUSY 204. Lorsque le signal BUSY 204 est positionné, c'est-à-dire est actif, le signal de mise à jour 212 devient inactif et par conséquent la mise à jour du registre 136 est arrêtée. L'entrée de positionnement SET 205 reçoit un signal de sortie décodé de la part du décodeur 202, qui reçoit des signaux d'entrée MISS_CAV 101 et TRANS-TYPE 201. Le signal TRANS_TYPE 201 peut être l'un quelconque des signaux suivants, sans y être limité, à savoir un "insuccès de charge résultant d'une instruction de lecture", un insuccès "de mémorisation" résultant d'une instruction d'écriture, et une réponse de contrôle de commande cohérence. Le signal TRANS_TYPE 201 est dérivé de l'instruction actuellement traitée par le mécanisme 39b de traitement d'instructions (figure 1) et à partir de signaux reçus de la part de la commande 102 d'interface du système (figure 1).

Le décodeur 202 délivre un signal de positionnement SET 205 actif lorsque le signal MISS_CAV 101 indique une adresse valable présente dans l'adresse MISS_COPY_IN ADDR 104, et lorsque le signal TRANS_TYPE 201 indique que la transaction est traitée dans l'interface transactionnelle entre le mécanisme 39b de traitement d'instructions (figure 1) et la commande 102 d'interface du système (figure 1) est une demande d'accès en mémoire résultant d'un insuccès en mémoire pour l'une quelconque des fentes ARBSLOT 48 ou bien résultant d'un contrôle de cohérence d'antémémoire, qui sera décrit plus loin de façon plus détaillée.

Le registre 136 met à jour continûment son contenu tant que le signal BUSY 204 reste inactif (c'est-à-dire lorsque le signal de mise à jour 212 est actif). Lorsque le signal BUSY 204 devient actif, le registre 136 arrête la mise à jour de son contenu et la fente DPRESLOT 200 envoie un signal ACCESS_REQ (représenté sur la figure 1) qui présente le contenu actuel de l'indice CACHE INDEX 136b, du signal TAG 136c et du signal STORE 136d présents respectivement au niveau de ACCESS_ADDR 114, de ACCESS_TAG 116 et de ACCESS_STORE 218, à l'antémémoire DCACHE 24.

Dans le cas où il existe un succès potentiel dans l'antémémoire DCACHE 24, l'indicateur d'état 82 reproduit une ou des lignes valables dans l'antémémoire comme cela est décrit de façon plus détaillée dans le brevet US 5 758 178. En outre le mécanisme 108 de comparaison d'étiquette lit la ou les étiquettes DCACHE_TAG 81 et les compare à l'étiquette ACCESS_TAG 116 associée à l'adresse d'accès ACCESS_ADDR 114. Lorsqu'il existe une concordance, le mécanisme 108 de comparaison d'étiquette en conclut qu'il existe un insuccès et supprime le signal ~HIT 118 pour indiquer un succès, ce qui a pour effet que l'entrée CLR 206 du circuit de verrouillage d'occupation 203 est activée, ce qui a pour effet que le signal BUSY 204 est supprimé.

Lorsque l'accès à l'antémémoire conduit à un insuccès sur la base d'un indicateur d'état 82 ou sinon lorsque l'étiquette DCACHE_TAG 81 ne concorde pas avec l'étiquette ACCESS_TAG 116, alors le mécanisme 108 de comparaison d'étiquettes active le signal ~HIT 118 pour indiquer un insuccès. Un mécanisme comparateur 55 reçoit un indice de l'antémémoire de la part de MISS_COPY IN ADDR 104, comme indiqué par la flèche de référence 146, et le compare à l'indice CACHE INDEX 136b provenant du registre 136, comme indiqué par la flèche de référence 147. Les résultats du mécanisme de comparaison 145 sont transférés à

une porte ET 214, comme indiqué par la flèche de référence 249. Pourvu que le signal de commande d'insuccès MISS_CAV 101 soit activé, le signal de comparaison 149 peut ramener à l'état initial le circuit de verrouillage d'occupation 203 ce qui provoque la suppression du signal BUSY 204. Dans cette forme de réalisation donnée à titre d'exemple, le signal de comparaison 149 permet la reprise de la mise à jour du registre 136 après que le signal MISS_GRANTED 112 a et reçu par la fente DPRESLOT 200.

Le circuit de verrouillage d'occupation 203 peut être également ramené à l'état initial lorsqu'il existe déjà une demande en cours pour la ligne de l'antémémoire. Si l'une quelconque des fentes ARBSLOT 148 a déjà demandé la même ligne d'antémémoire à partir du système hiérarchique de mémoire (non représenté), alors la fente DPRESLOT 200 est avertie par l'indicateur d'état 82 (figure 1), lorsque l'indicateur d'état 82 indique un état d'insuccès en cours comme cela est décrit de façon plus détaillée dans le brevet US N°5 758 178. Dans ce cas, le mécanisme de comparaison d'étiquette 108 active le signal HIT_DM 121 (comme représenté sur la figure 1), qui est envoyé, conjointement avec un signal ACCESS_+_2, désigné par le chiffre de référence 158 et représentant deux cycles après le signal ACCESS (figure 3), à la porte logique ET 211, ce qui a pour effet de supprimer le signal BUSY 204.

Un autre cas, lors duquel le circuit de verrouillage BUSY 203 peut être effacé se présente lorsqu'un signal indicatif de l'apparition d'un événement catastrophique inattendu est reçu de la part de l'entrée 208 de la porte logique OU 207. Un événement catastrophique inattendu peut être par exemple une interruption de l'unité centrale CPU.

Etant donné que le signal BUSY 204 est envoyé à la porte logique ET 207, lorsqu'il est inactif, c'est-à-dire dans un état désactivé, la fente DPRESLOT 200 ne peut

pas formuler une demande MISS_REQUEST 211. Le signal BUSY désactivé 204 amène également le registre 136 à reprendre la mise à jour de son contenu.

Si d'autre part la demande ACCESS_REQ 115 de cette ligne adjacente de l'antémémoire a abouti à un insuccès, c'est-à-dire qu'un signal ~HIT 139 et le signal BUSY 204 sont activés, alors la porte logique ET 137 délivre le signal de demande conduisant à un insuccès MISS_REQUEST 111 au dispositif d'arbitrage en cas d'insuccès 107 (figure 1). Le dispositif d'arbitrage en cas d'insuccès 107 effectue un arbitrage en donnant la priorité aux différentes demandes conduisant à un insuccès, qui peuvent être produites par les différentes fentes ARBSLOT 48 et/ou la fente DPRESLOT 200. Eventuellement le dispositif d'arbitrage en cas d'insuccès 107 délivre un signal MISS_GRANTED 112 pour accorder la demande conduisant à un insuccès. Ce signal est envoyé à l'étage d'attaque 213 dans la fente DPRESLOT 200, qui à son tour active le signal de commande conduisant à un insuccès MISS_CAV 101 envoyé à la commande 102 d'interface du système. La commande 102 d'interface du système envoie à son tour une demande de mémoire au système hiérarchique de mémoire (non représenté) pour la ligne de transmission de données sur la base de l'adresse MISS_COPY_IN ADDR 104.

La figure 2A représente un schéma-bloc pris à titre d'exemple des parties importantes de la logique d'arbitrage des ports d'antémémoire en fonction d'une forme de réalisation préférée de la présente invention, dans laquelle trois étages d'attaque 220 sont ajoutés, chacun d'eux est validé, c'est-à-dire est autorisé à délivrer le signal présent aux entrées respectives, lorsque le signal CACHE_GRANTED 221 est activé par le dispositif 222 d'arbitrage des ports de l'antémémoire, qui peut faire partie de l'antémémoire DCACHE 24. Le signal CACHE-GRANT 221 est activé lors d'une réception, et d'un arbitrage, du signal CACHE_REQ 223 qui est reçu de la part de la porte logique

ET 224. La porte logique ET 224 reçoit à son tour, au niveau de ses entrées, l'impulsion d'horloge 225, le signal BUSY 204, le signal ~ACCESS+1 (c'est-à-dire le complément d'un cycle d'horloge après le signal ACCESS_REQ 115) 226 et un signal ~ACCESS+2 (c'est-à-dire le complément de deux cycles d'horloge après le signal ACCESS_REQ 115) 158.

On va maintenant décrire le procédé de l'opération de pré-extraction selon l'invention en se référant à un organigramme indiqué à titre d'exemple et représenté sur la figure 3. Lors du pas 301, l'interface transactionnelle entre le mécanisme 39b de traitement d'instructions et la commande 102 d'interface du système (qui sera désignée simplement ci-après sous l'expression "interface transactionnelle") est contrôlée en permanence pour détecter la présence éventuelle d'une quelconque transaction, ce qui peut être réalisé par exemple par contrôle d'une activation du signal MISS_CAV 101 dans la fente DPRESLOT 200 représentée sur la figure 2.

Lorsqu'une transaction est détectée, une détermination est faite, lors du pas 302, pour savoir s'il existe une adresse valable présente dans l'interface transactionnelle. Dans l'exemple représenté sur la figure 2, on peut supposer qu'une adresse valable est présente, par exemple lorsque le signal MISS_CAV 101 est activé. Lorsqu'il est établi qu'une adresse valable n'est pas présente au niveau de l'interface de transaction, alors le processus revient au pas 301, c'est-à-dire que le contrôle de l'interface transactionnelle continue.

D'autre part, si une adresse valable est détecté, la procédure passe au pas 303, lors duquel une détermination est faite pour savoir si la transaction est une demande d'accès en mémoire résultant d'un insuccès dans l'antémémoire. Dans l'exemple de la figure 2, cette détermination peut être faite sur la base du signal TRANS_TYPE 201. Lorsqu'il est établi que la transaction

n'est pas un insuccès dans l'antémémoire, alors la procédure revient au pas 301, c'est-à-dire que le contrôle de l'interface transactionnelle se poursuit.

Cependant, si la transaction est une demande d'accès en mémoire résultant d'un insuccès dans l'antémémoire, alors lors du pas 304, le contrôle de l'interface transactionnelle est bloqué. Dans la fente DPRESLOT 200 par exemple, la mise à jour du registre 136 est arrêtée par positionnement du circuit de verrouillage d'occupation 203. Ensuite lors du pas 305, une ou plusieurs adresses de lignes de transmission de données devant être pré-extraites sont calculées. Par exemple dans la fente DPRESLOT 200, la logique d'adresse adjacente 213 calcule les adresses devant être pré-extraites par inversion d'un ou de plusieurs bits (par exemple le bit le moins significatif (LSB)) de l'adresse de la ligne de transmission de données, l'accès tenté à ce bit ayant provoqué l'insuccès dans l'antémémoire, présent dans MISS/COPY_IN ADDR 104.

Lors du pas 306, une opération de consultation dans l'antémémoire est exécutée pour les adresses calculées pendant le pas 305 indiqué ci-dessus. Par exemple dans l'exemple de la figure 2, la fente DPRESLOT 200 délivre une demande ACCESS_REQ 115 présentant le contenu actuel du signal CACHE INDEX 136b, du signal TAG 136c et du signal STORE 136d respectivement dans ACCESS_ADDR 114, ACCESS_TAG 116 et ACCESS_STORE 218, à l'antémémoire DCACHE 24.

Lors du pas 307, le résultat de l'opération de consultation d'antémémoire est examiné pour déterminer si les lignes de transmission de données devant être pré-extraites sont déjà présentes dans l'antémémoire, c'est-à-dire s'il se produit un insuccès dans l'antémémoire. Par exemple, dans l'exemple de la figure 2, la fente DPRESLOT 200 détermine qu'un insuccès dans l'antémémoire est apparu en observant le fait que le signal ~HIT 118 est désactivé par le mécanisme 108 de comparaison d'étiquette. Si un

insuccès dans l'antémémoire est apparu, la procédure revient au pas 301, et le contrôle de l'interface transactionnelle reprend.

Si cependant, lors du pas 307, un insuccès dans l'antémémoire est détecté, la procédure passe au pas 308, lors duquel une détermination est faite pour savoir si une demande de la ou des lignes de transmission de données devant être pré-extraites est déjà effectuée, par exemple par une fente ARBSLOT 48 dans l'exemple représenté sur la figure 2. Dans l'exemple de la figure 2, une demande en cours pour la ligne de transmission de données peut être détectée à partir du signal HIT_DM 121. S'il est établi qu'une demande de la ligne de transmission de données est déjà en cours, alors la procédure revient au pas 301, et le contrôle de l'interface transactionnelle reprend.

Enfin, lors du pas 309, si aucune demande antérieure pour la ligne de transmission de données n'est en cours, une demande pour la ligne de transmission de données devant être extraite est délivrée, par exemple par délivrance du signal MISS_REQUEST 111 dans l'exemple de la figure 2, qui éventuellement conduit au fait que le signal MISS_CAV 101 est activé, et provoque un accès du système hiérarchique de mémoire pour la ou les lignes de transmission de données. Dans une forme de réalisation préférée, une fois que la demande concernant la ou les lignes de transmission de données devant être pré-extraites est délivrée (MISS_CAV 101 activé), la procédure revient immédiatement au pas 301 et l'ensemble de la procédure est répété continûment. Sur la figure 2 par exemple la commande 102 de l'interface du système traite avantageusement l'accès actuel du système hiérarchique de mémoire en permettant à la fente DPRESLOT 200 de continuer la procédure décrite plus haut. Lorsque l'adresse de la ligne de transmission de données devant être pré-extraite est positionnée sur le MISS/COPY_IN ADDR 104 en tant que partie de la

demande de l'interface 102 de commande du système, le comparateur 145 reçoit un indice d'antémémoire identique à ses deux entrées 146 et 147, et par conséquent le signal BUSY 204 est désactivé, ce qui a pour effet que le registre 136 reprend la mise à jour de son contenu.

Si une demande conduisant à un insuccès est déclenchée par une instruction dans la fente ARBSLOT 48, une demande qui concorde avec l'adresse présente à l'entrée de comparaison 147 avant que la fente DPRESLOT 200 reçoive le signal MISS_GRANTED 112, le signal BUSY 204 est désactivé, et la mise à jour du registre 136 reprend.

On va décrire ci-après le système et le procédé de prévidage de l'antémémoire selon l'invention conformément aux principes de la présente invention, en référence à des exemples de formes de réalisation illustrées sur les figures 4 et 5.

Conformément à une forme de réalisation préférée selon la présente invention, une ou plusieurs fentes de contrôle de cohérence de l'antémémoire (CCCSLOT) sont ajoutées au mécanisme 39b de traitement d'instructions (figure 1). Sinon, une ou plusieurs des fentes ARBSLOT représentées sur la figure 1 peuvent être modifiées de manière à assumer les fonctions de la fente CCCSLOT qui vont maintenant être décrites de façon plus détaillée.

En particulier la figure 4 représente un schéma-bloc d'une forme de réalisation prise à titre d'exemple de la fente de contrôle de cohérence de l'antémémoire (CCCSLOT), qui peut apparaître et fonctionne d'une manière très semblable à une fente ARBSLOT 48 comme cela est décrit dans le brevet US N°5 758 178, avec comme différences clés notamment l'addition du circuit de verrouillage de tâche exécutée '02 et l'étage d'attaque 407, et le fait que l'adresse 128 et l'étiquette 134 sont reçues non pas du calculateur d'adresses 58, mais de la commande 102 de l'interface du système.

Lorsque l'un quelconque des multiples processeurs dans un système de calcul à processeurs multiples demande une ou plusieurs lignes de transmission de données à partir de la hiérarchie de la mémoire, la demande de mémoire et la ou les adresses de la ou de la pluralité de lignes de transmission de données apparaissent dans l'interface 22 du système (figure 1). L'interface 102 du système de chaque processeur délivre, lors de la détection de la demande de mémoire, un signal CCC_INSERT 401 à son mécanisme 39B de traitement des instructions.

A cet effet, dans la forme de réalisation préférée de la présente invention, la fente respective CCCSLOT 400 de chacun des processeurs reçoit le signal ADDR 128, le signal TAG 134 et le signal CCC_INSERT 401 de la part de la commande 102 de l'interface du système, le signal ADDR 128 et le signal TAG 134 étant associés à la ligne de transmission de données qui est demandée par un autre processeur dans le système. Le signal CCC_INSERT 401 est utilisé en tant que signal d'horloge envoyé au registre 136 de la fente CCCSLOT 400, ce qui permet au registre 136 de mettre à jour son indice CACHE INDEX 136b et l'étiquette TAG 136c avec respectivement le signal ADDR 128 et le signal TAG 134. Le signal CCC_INSERT 401 est également envoyé à l'entrée d'effacement (CLR) du circuit de verrouillage de travail exécuté 402, qui peut être par exemple une bascule bistable à positionnement et remise à l'état initial (S-R). Lorsque le signal d'entrée CLR est reçu, le signal de sortie du circuit de verrouillage de travail exécuté 402 devient inactif. L'inverseur 410 inverse le signal de sortie du circuit de verrouillage de travail exécuté 401, ce qui provoque l'envoi d'un signal actif ~DONE à l'entrée de la porte logique ET 137 comme représenté.

Lors de la réception du signal CCC_INSERT 401, la fente CCCSLOT 400 délivre un signal ACCESS_REQ 115 à

l'antémémoire DCACHE 24 et applique le signal CACHE INDEX 136b et le signal TAG 134 respectivement à ACCESS_ADDR 114 et ACCESS_TAG 116. En réponse, l'antémémoire DCACHE 24 délivre le ou les signaux DCACHE TAG 81 et STATUS 82 comme représenté sur la figure 1. D'une manière très similaire à ce qui est décrit précédemment dans le brevet N°5 758 178 en rapport avec la fente ARBSLOT 48, un signal MISS_REQUEST 111 est produit lorsque la ligne de transmission de données correspondant au signal ADDR 128 et au signal TAG 134 est absente de l'antémémoire DCACHE 24, et lorsqu'aucune autre demande pour la même ligne de transmission de données n'est en cours. Lorsque le MISS_ARBITRATOR 107 renvoie le signal MISS_GRANTED 112 en réponse au signal MISS_REQUEST 111, le signal MISS_GRANTED 112 est envoyé à l'entrée de positionnement SET du circuit de verrouillage de travail exécuté 402 en produisant ainsi un signal DONE actif servant à empêcher la délivrance d'un quelconque autre signal MISS_REQUEST 111.

Le signal MISS_GRANTED 112 autorise également l'étage d'attaque 407 à transmettre le contenu actuel du signal ~HIT 136a du registre 136 au signal CCC_MISS/HIT 408, qui est envoyé à la commande 102 d'interface du système. Sur la base du signal CCC_MISS/HIT reçu 408 et du signal STATUS 82, la commande 102 de l'interface du système détermine si la réécriture ou le vidage de la ligne de transmission de données (c'est-à-dire celle désignée par MISS/COPY_IN ADDR 104) depuis l'antémémoire DCACHE 24 au système hiérarchique de mémoire (non représenté) est requis. Dans une forme de réalisation de la présente invention, chaque fois que la ligne de transmission de données trouvée dans la mémoire DCACHE 24, c'est-à-dire si le signal CCC_MISS/HIT 408 est actif et que le signal STATUS 82 indique que la ligne de l'antémémoire est modifiée, la commande 102 de l'interface du système amène la ligne de transmission de données (c'est-à-dire pointée par

MISS_COPY_IN ADDR 104) à être écrite à l'extérieur du processeur qui a demandé la ligne de l'antémémoire.

Lorsque la fente DPRESLOT 200 reçoit le résultat du contrôle de cohérence de l'antémémoire indiquée à l'entrée 201 du signal TRANS_TYPE, qui est piloté par la fente CCCSLOT 400, la fente DPRESLOT 200 déclenche une opération de prévidage conformément aux principes de la présente invention, qui va être décrite ci-après en référence aux figures 2 et 5.

En particulier la figure 5 représente un organigramme d'un mode de mise en oeuvre pris à titre d'exemple du processus de prévidage, lors du pas 501 duquel l'interface de transaction entre le mécanisme 39b de traitement d'instructions et la commande 102 de l'interface du système (qui sera désignée ci-après simplement par "interface transactionnelle") est contrôlée en permanence pour déterminer l'existence éventuelle d'une transaction, ce qui peut être réalisé par exemple par contrôle d'une activation du signal MISS_CAV 101 dans la fente prise à titre d'exemple DPRESLOT 200 représentée sur la figure 2.

Lorsqu'une transaction est détectée, une détermination est faite pour savoir s'il existe une adresse valable présente dans l'interface de transaction, et ce par exemple par détection du signal MISS_CAV 101 activé (pas 502). Lorsqu'il est établi qu'une adresse valable n'existe pas dans l'interface transactionnelle, alors la procédure revient au pas 501, c'est-à-dire que le contrôle de l'interface de transaction se poursuit.

D'autre part, si une adresse valable est détectée, la procédure passe au pas 503, pendant lequel une détermination est faite pour savoir si la transaction est une réponse de cohérence résultant de contrôle de cohérence de l'antémémoire. Lorsqu'il est établi que la transaction n'est pas une réponse de cohérence de l'antémémoire, alors le processus revient au pas 501, c'est-à-dire que le

contrôle de l'interface transactionnelle se poursuit.

Cependant si la transaction est une réponse de cohérence, par exemple une transaction de réponse de cohérence requérant la copie de données modifiées comme indiqué par le signal STATUS 82, le contrôle de l'interface de transaction est arrêté, par exemple par positionnement du circuit de verrouillage d'occupation 203 pour arrêter la mise à jour du registre 136. Ensuite lors du pas 505, une ou plusieurs adresses de lignes de transmission de données devant être pré-extraites sont calculées. La logique d'adresses adjacentes 213 calcule les adresses devant être pré-extraites, par inversion d'un ou de plusieurs bits (par exemple le bit le moins significatif (LSB)) de l'adresse de la ligne de transmission de données présente dans MISS/COPY_IN ADDR 104.

Lors du pas 506, une opération de consultation dans l'antémémoire est exécutée pour les adresses calculées pendant l'étape 105 indiquée précédemment. La fente DPRESLOT 200 délivre un signal ACCESS_REQ 115 qui envoie les contenus actuels de l'indice des signaux CACHE INDEX 136b, TAG 136c et STORE 136d respectivement dans les lignes ACCESS_ADDR 114, ACCESS_TAG 116 et ACCESS_STORE 218, à l'antémémoire DCACHE 24.

Lors du pas 507, le résultat de l'opération de consultation de l'antémémoire est examiné pour déterminer si la ou les lignes de transmission de données devant être pré-extraites sont présentes dans l'antémémoire, c'est-à-dire que la fente DPRESLOT 200 détermine qu'un insuccès dans l'antémémoire est apparu en observant le fait que le signal ~HIT 118 est désactivé par le mécanisme 108 de comparaison d'étiquette. Si un insuccès dans l'antémémoire est apparu, la procédure revient au pas 501 et le contrôle de l'interface transactionnelle reprend.

Si cependant lors du pas 507, un insuccès dans l'antémémoire est détecté, la procédure passe au pas 508,

lors duquel une détermination, établissant si une demande pour la ou les lignes de transmission de données devant être prévidées est déjà effectuée, par exemple par une fente ARBSLOT 48 représentée sur la figure 1, moyennant l'observation d'un signal HIT_DM 121. S'il est établi qu'une demande pour la ligne de transmission de données est déjà en cours, alors la procédure revient au pas 501 et le contrôle de l'interface transactionnelle reprend.

Enfin, lors du pas 509, si aucune demande antérieure pour la ligne de transmission de données n'est en cours, une transaction de vidage pour la ligne de transmission de données devant être prévidée est produite par exemple par la délivrance de MISS_REQUEST 111, qui permet un accès du système hiérarchique de mémoire d'écrire, sous l'action de la commande 102 d'interface du système, les lignes de transmission de données de l'antémémoire DCACHE 24 au système hiérarchique de mémoire. A cet effet, le signal d'entrée ~HIT envoyé à la porte logique ET 137 peut être inversé de manière à permettre l'utilisation de la fente DPRESLOT 200 pour une opération de prévidage, par exemple lorsque le signal TRANS_TYPE 201 indique un contrôle de cohérence de mémoire. Dans une forme de réalisation préférée de la présente invention, l'état STATUS 82 est consulté, et la ligne de transmission de données devant être prévidée est vidée uniquement si l'état de la ligne de transmission de données devant être prévidée indique que les données sont modifiées. Sinon, la ligne de transmission de données devant être prévidée peut être vidée indépendamment de son état. Dans un mode de mise en oeuvre préféré, une fois que la demande de la ou des lignes de transmission de données devant être prévidées sont délivrées, le processus revient immédiatement au pas 501 et l'ensemble du processus est répété continûment.

Comme on peut le noter, on a décrit un système efficace de pré-extraction et/ou de prévidage d'une ou

plusieurs lignes de transmission de données, qui n'affecte pas les autres composants et par conséquent peut être aisément intégré dans un système de traitement non ordonnée, et qui réduit également de multiples demandes de mémoire redondantes.

Bien que l'invention ait et décrite en référence à des formes de réalisation prise à titre d'exemples, les spécialistes de la technique sont à même d'apporter différentes modifications aux formes de réalisation décrites de l'invention sans sortir du cadre de cette dernière. Les termes et les descriptions utilisés sont donnés uniquement à titre d'illustration et ne doivent pas être considérés comme des limitations. En particulier, bien que le procédé de la présente invention ait et décrit sur la base d'exemples, les étapes du procédé peuvent être exécutés dans un ordre différent de celui représenté ou bien simultanément. Le spécialiste de la technique notera que ces variantes et d'autres variantes sont possibles sans sortir du cadre de l'invention.

LEGENDES DES FIGURES

Figure 1 :

108. Comparaison d'étiquettes

107. Dispositif d'arbitrage en cas d'insuccès

135. Logique formant indicateur de DM

102. Commande d'interface du système

22. Interface du système

a. Données

b. Registres de renommage de files d'attente MQUEUE

c. Calculateur d'additions

d. Mécanisme de traitement d'instructions

e. Autres ARBSLOT 48

f. Art antérieur

Figure 2 :

213. Logique d'adresse adjacente

237, 214, 215, 211. ET

202. Décodage

207. OU

145. Comparateur

Figure 2a :

222. Dispositif d'arbitrage de ports de l'antémémoire

Figure 3 :

301. Contrôle de transactions

302. Adresse valable ?

303. TRANS_TYPE = insuccès dans l'antémémoire

304. Interruption de transaction de contrôle

305. Calcul d'adresse(s) adjacente(s) & d'étiquettes

306. Consultation de l'antémémoire

307. Succès ?

308. Demande conduisant à un insuccès en cours ?

309. Délivrance de miss_request

Sur cette figure remplacer YES par OUI et NO par NON.

Figure 4 :

137, 148, 162. ET

a. En provenance de la commande 102 de l'interface du système

b. En provenance de la commande 102 de l'interface du système

Figure 5 :

501. Contrôle de transactions
502. Adresse valable ?
503. TRANS_TYPE = contrôle de cohérence ?
504. Interruption de transaction de contrôle
505. Calcul d'adresse(s) adjacente(s) & d'étiquettes
506. Consultation de l'antémémoire
507. Succès ?
508. Demande conduisant à un insuccès en cours ?
509. Délivrance de miss_request

Sur cette figure remplacer YES par OUI et NO par NON.

REVENDICATIONS

1. Dispositif (100) pour réduire le temps de latence dans un système d'exécution non ordonnée d'instructions, caractérisé en ce qu'il comprend :

une pluralité de fentes (48) de tampons de réarrangement d'adresses, dans chacune desquelles est mémorisée une adresse correspondant à une ligne de transmission de données d'une pluralité de lignes de transmission de données, chacune de ladite pluralité de lignes de transmission de données étant mémorisée dans au moins une antémémoire (24) et un système hiérarchique de mémoire;

au moins une fente de pré-extraction (200) configurée pour, lors de la détection d'un insuccès d'une première ligne de transmission de données dans l'antémémoire, résultant d'une demande d'accès provenant d'au moins une fente de ladite pluralité de fentes (48) de tampons de réarrangement d'adresses, déterminer au moins une ligne additionnelle de transmission de données devant être pré-extraite dans ladite antémémoire (24) à partir dudit système hiérarchique de mémoire; et

une logique associée à ladite au moins une fente de pré-extraction, ladite logique étant configurée de manière à fournir une indication du fait que ladite au moins une ligne additionnelle de transmission de données est déjà demandée à partir dudit système hiérarchique de mémoire par l'une quelconque de ladite pluralité de fentes (48) de tampons de réarrangement d'adresses.

2. Dispositif (100) pour réduire le temps de latence en cas d'insuccès dans une antémémoire, selon la revendication 1, caractérisé en ce que ladite au moins une fente de pré-extraction (200) comprend une logique d'adresses adjacentes (213) configurée de manière à délivrer une ou plusieurs adresses supplémentaires correspondant à ladite au moins une ligne additionnelle de transmission de données, ladite au moins une ligne additionnelle

de transmission de données possédant un emplacement de mémoire adjacent à ladite première ligne de transmission de données.

3. Dispositif (100) pour réduire le temps de latence en cas d'insuccès dans une antémémoire selon la revendication 2, caractérisé en ce que ladite logique d'adresses adjacentes (213) reçoit une première adresse correspondant à ladite première ligne de transmission de données et délivre une ou plusieurs adresses additionnelles par inversion d'un ou de plusieurs bits dans ladite première adresse.

4. Dispositif (100) pour réduire le temps de latence en cas d'insuccès dans une antémémoire selon la revendication 2, caractérisé en ce qu'il comporte en outre :

un circuit de verrouillage d'occupation (203) comportant une entrée de positionnement (205) entre l'effacement (206), ledit circuit de verrouillage d'occupation (203) étant configuré de manière à délivrer un signal d'occupation (204), et ledit signal d'occupation (204) étant actif lorsque ladite entrée de positionnement (205) est déclenchée, et inactif lorsque ladite entrée d'effacement (206) est déclenchée; et

un registre (136) configuré de manière à mémoriser un indice d'antémémoire (136b) et une étiquette (136c), ces deux éléments étant dérivés d'une adresse reçue de la part de ladite logique d'adresses adjacentes (213), ledit registre (136) recevant ladite adresse de la part de ladite logique d'adresses adjacentes (213) lors de la réception d'un signal de mise à jour (212), ledit signal de mise à jour (212) étant produit par inversion dudit signal d'occupation (204).

5. Dispositif (100) pour réduire le temps de latence en cas d'insuccès dans une antémémoire selon la revendication 4, caractérisé en ce qu'il comporte en

outre : une logique de décodage (202) pour recevoir un signal valable d'adresse d'insuccès dans l'antémémoire (101) et un type de transaction (201), ladite logique de décodage (202) étant configurée de manière à déclencher ladite entrée de positionnement (205) dudit circuit de verrouillage d'occupation (203) lorsque ledit signal valable d'adresse d'insuccès dans l'antémémoire (101) introduit indique qu'une demande de transaction est faite audit système hiérarchique de mémoire, et que ledit type de transaction (201) introduit indique que ladite demande de transaction est effectuée en raison d'un insuccès d'accès dans l'antémémoire.

6. Procédé pour réduire le temps de latence en cas d'insuccès dans une antémémoire dans un système d'exécution non ordonnées d'instructions, caractérisé en ce qu'il comprend :

la détection de l'apparition d'un insuccès d'une première ligne de données dans l'antémémoire;

le calcul d'une adresse d'au moins une ligne additionnelle de transmission de données devant être préalablement extraites dans une antémémoire (24) à partir d'un système hiérarchique de mémoires; et

déterminer si une demande faite précédemment pour ladite au moins une ligne additionnelle de transmission de données provenant dudit système hiérarchique de mémoire est en cours.

7. Procédé pour réduire le temps de latence en cas d'insuccès dans l'antémémoire selon la revendication 6, caractérisé en ce que ladite étape de calcul de ladite adresse de ladite au moins une ligne additionnelle de transmission de données comprend :

l'inversion d'un ou de plusieurs bits d'une adresse de ladite première ligne de transmission de données.

8. Procédé pour réduire le temps de latence en

cas d'insuccès dans l'antémémoire selon la revendication 6, caractérisé en ce qu'il consiste en outre, dans le cas où ladite demande formulée précédemment est en cours, à empêcher qu'une demande de mémoire pour ladite au moins une ligne additionnelle de transmission de données soit à nouveau présentée.

9. Procédé pour réduire le temps de latence en cas d'insuccès dans l'antémémoire selon la revendication 8, caractérisé en ce qu'il consiste en outre, si ladite demande formulée précédemment n'est pas en cours, à délivrer une demande pour ladite au moins une ligne additionnelle de transmission de données à partir dudit système hiérarchique de mémoire.

10. Procédé pour réduire le temps de latence en cas d'insuccès dans l'antémémoire selon la revendication 8, caractérisé en ce qu'il consiste en outre à :

déterminer si ladite au moins une ligne additionnelle de transmission de données est présente dans ladite antémémoire (24);

si ladite au moins une ligne additionnelle de transmission de données est présente dans ladite antémémoire (24), empêcher qu'une demande de mémoire pour ladite au moins une ligne additionnelle de transmission de données soit à nouveau formulée; et

si ladite au moins une ligne additionnelle de transmission de données n'est pas présente dans ladite antémémoire (24), délivrer une demande pour ladite au moins une ligne additionnelle de transmission de données à partir dudit système hiérarchique de mémoire.

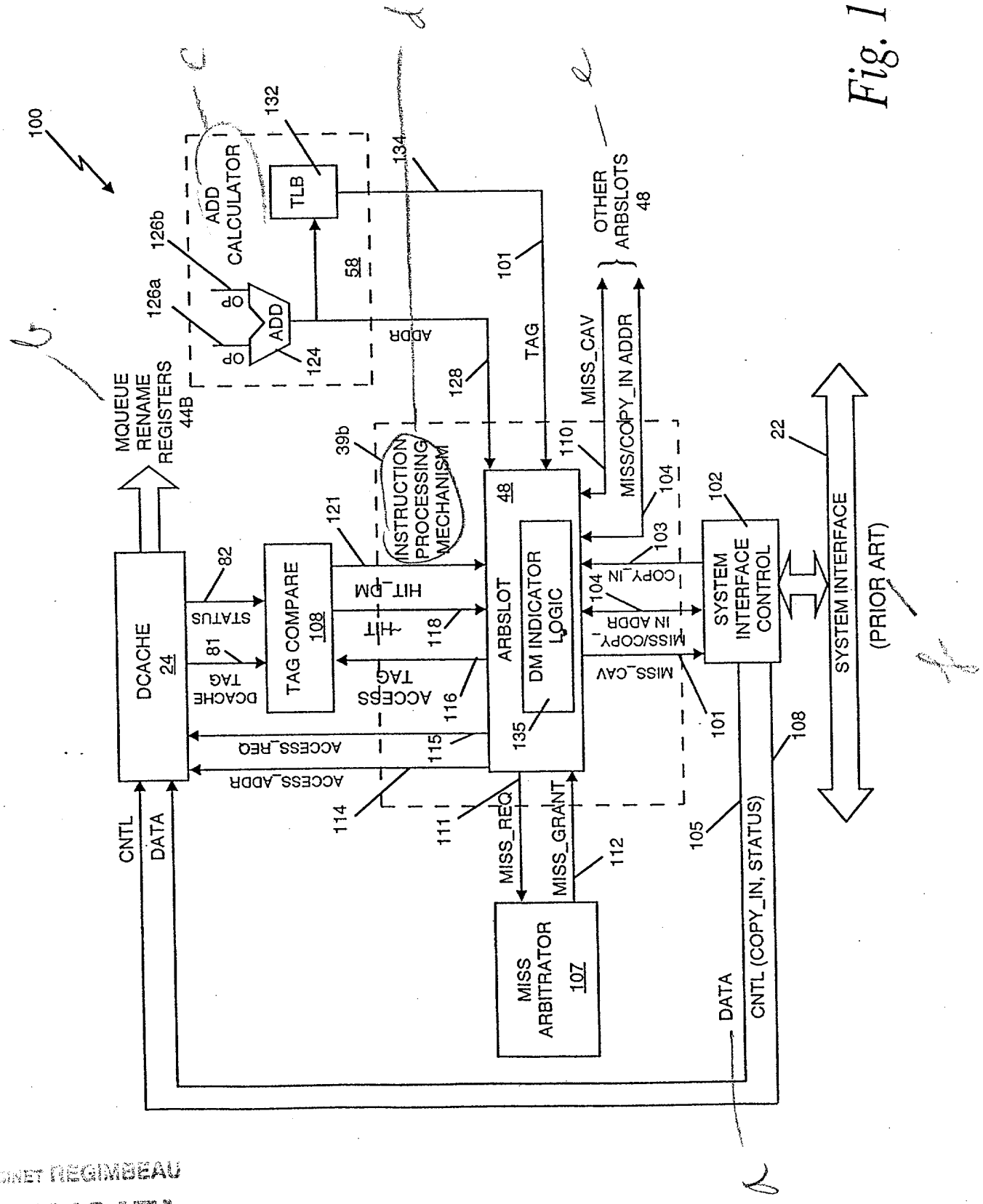


Fig. 1

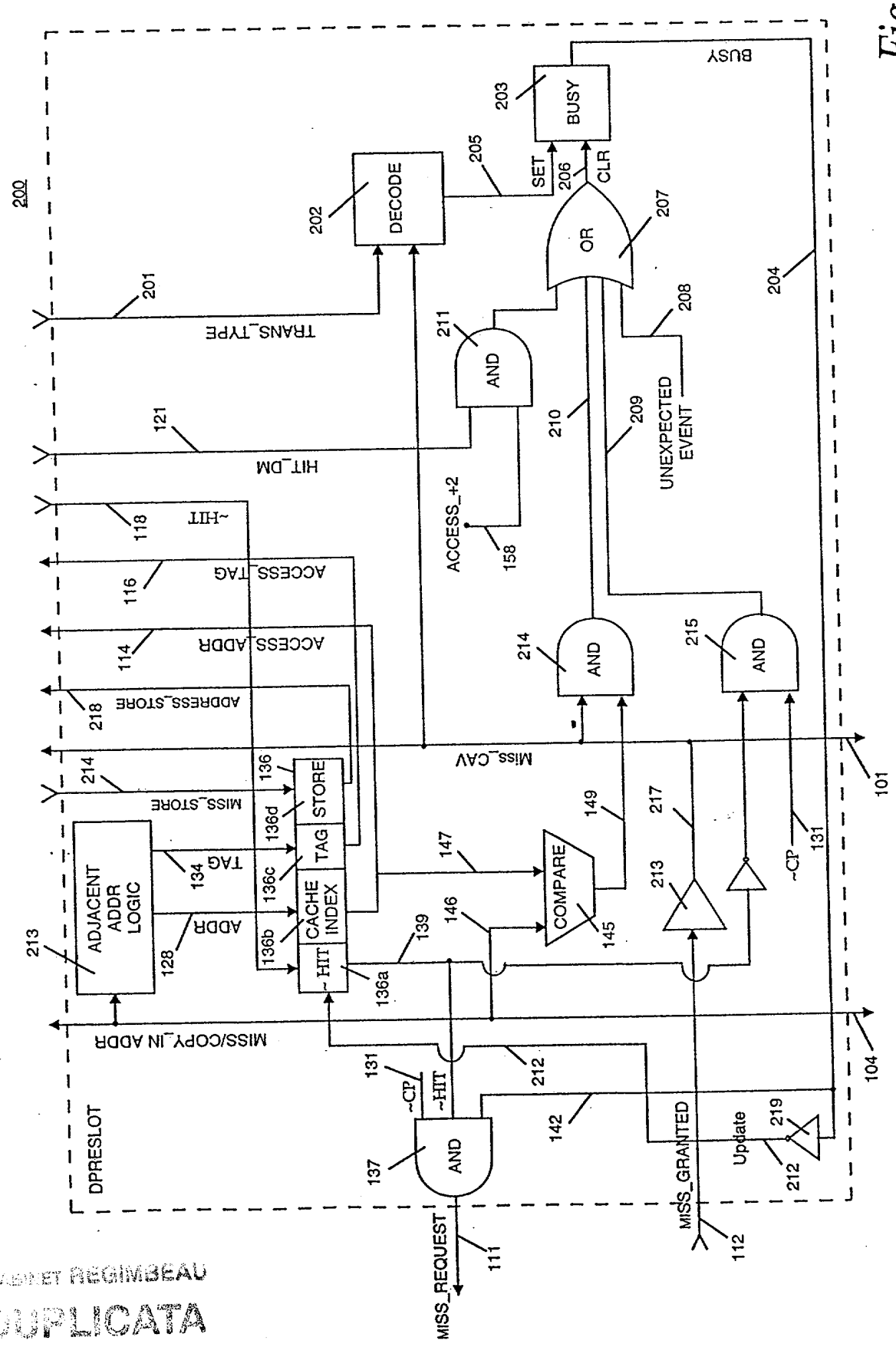


Fig. 2

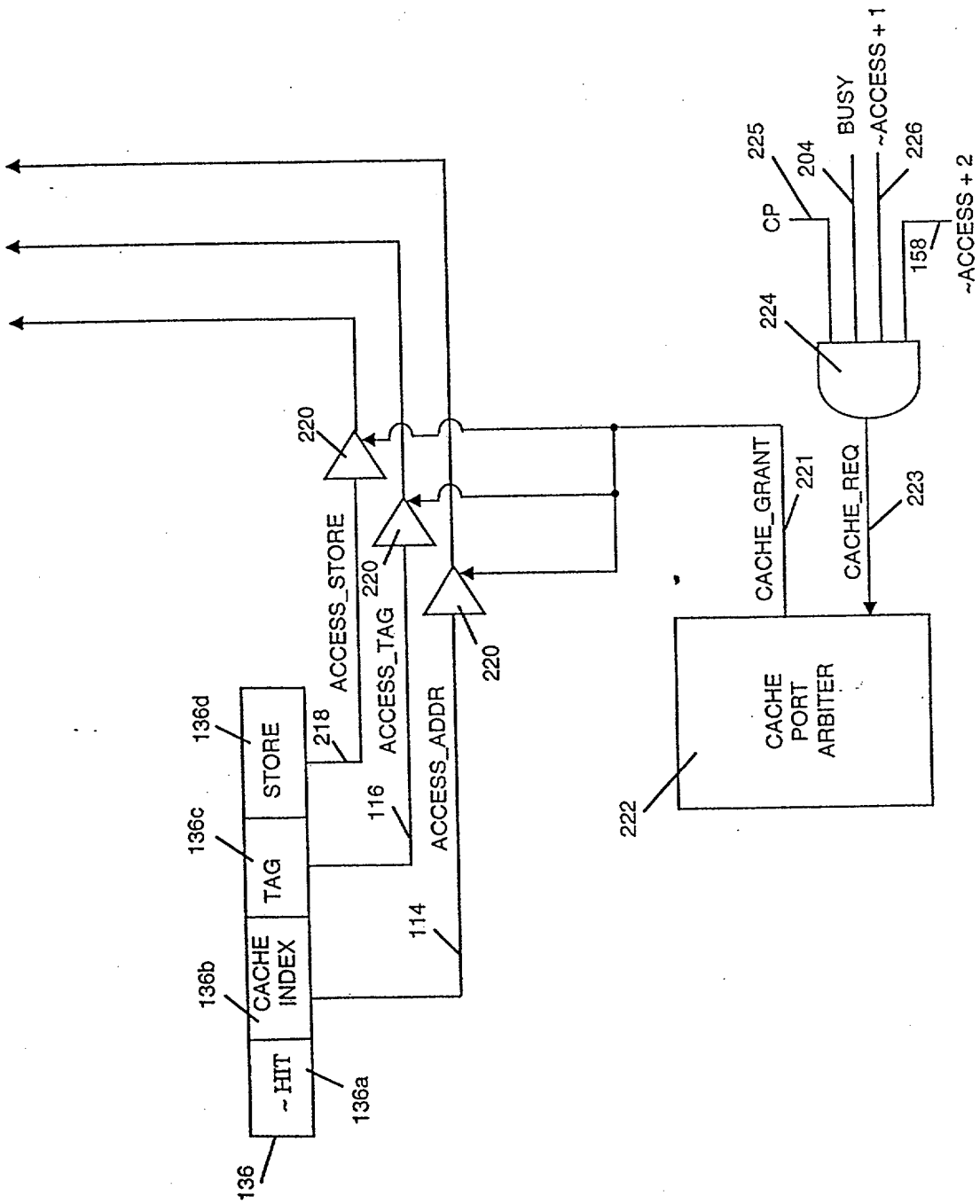


Fig. 2a

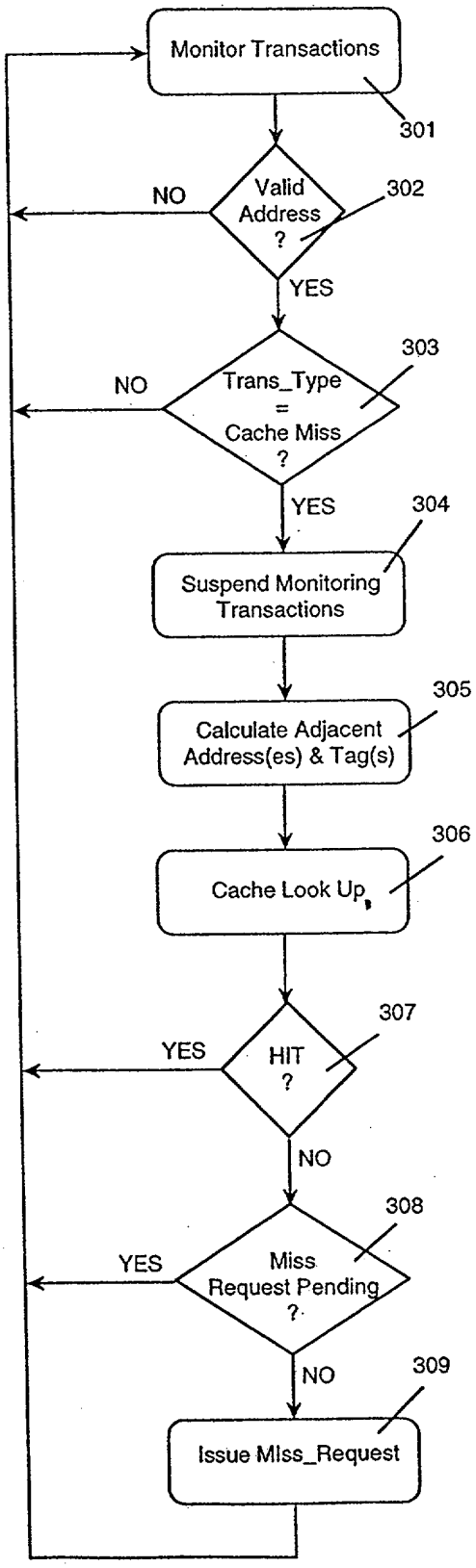


Fig. 3

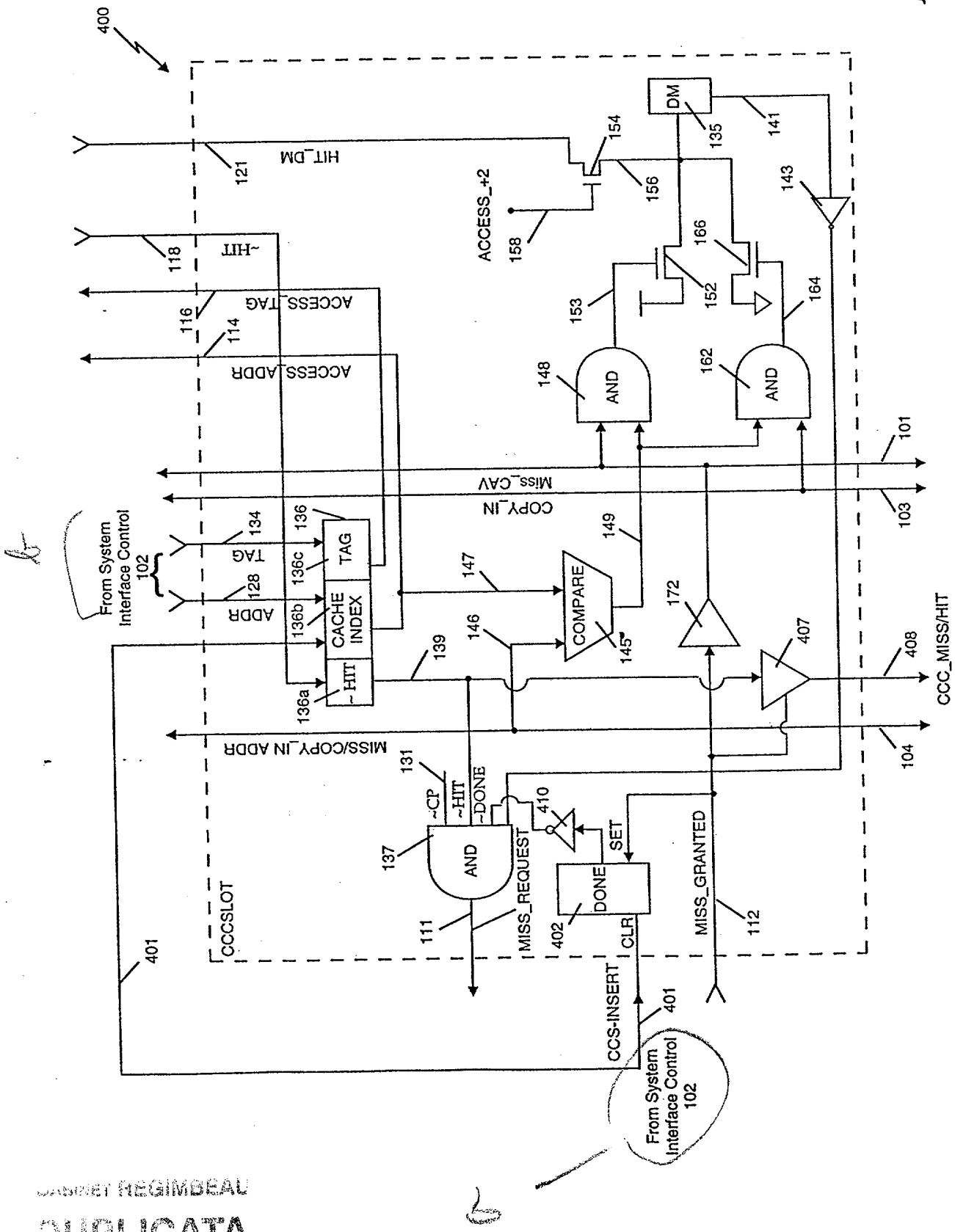


Fig. 4

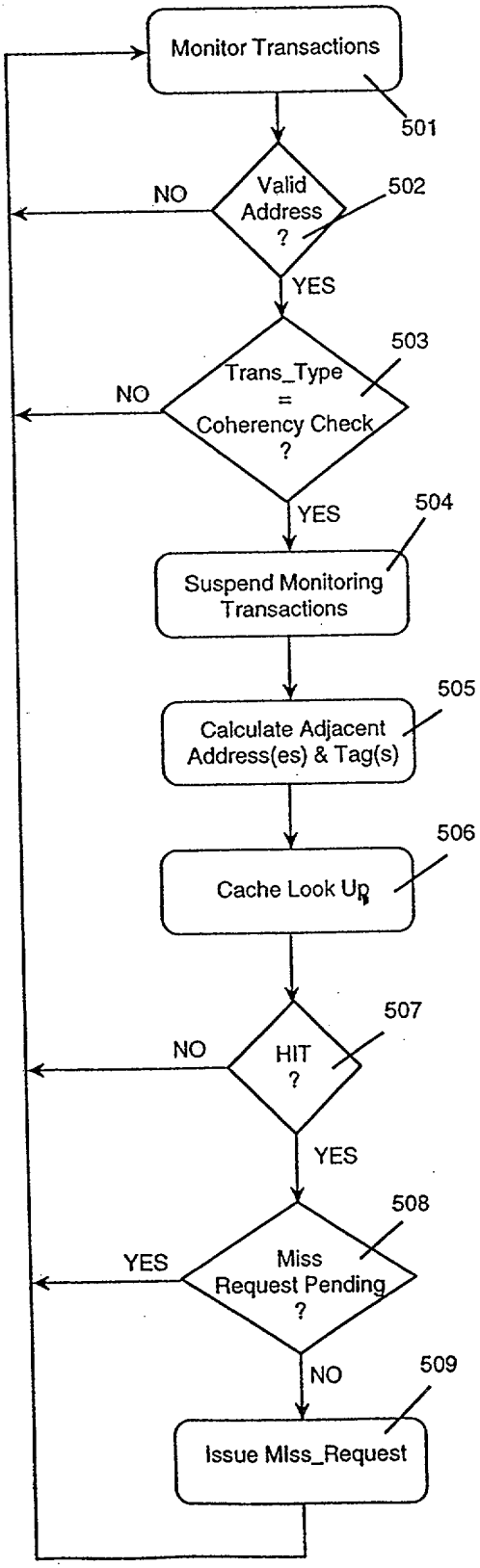


Fig. 5